IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

ISO 3297:2007 Certified ⁜ Impact Factor 8.102 ⁜ Vol. 12, Issue 4, April 2023

DOI: 10.17148/IJARCCE.2023.12449

PERFORMANCE EVALUATION OF CORRECT AND APPROXIMATE ADDERS USING CARRY-LOOKAHEAD AND CARRY SELECT ADDERS

Ms.J.Jasmin Shifa[1], S. Yogeshwari[2]

Assistant professor, Department of Electronics and Communication Engineering, Erode SenguntharEngineering College, Erode, Tamilnadu[1]

PG scholer, Department of Electronics and Communication Engineering, Erode SenguntharEngineering College, Erode, Tamilnadu[2]

Abstract: An adder is used to physically realise addition, which is a fundamental operation in microprocessing and digital signal processing technology. Two common high-speed, low-power adder architectures are the carry-lookahead adder (CLA) and the carry-select adder (CSLA). Using a hybrid CLA architecture, which substitutes a small-size ripple-carry adder (RCA) for a sub-CLA at the least significant bit positions, can increase the speed performance of a CLA architecture. On the other hand, by using binary-to-excess-1 code (BEC) converters, the power dissipation of a CSLA using full adders and 2:1 multiplexers can be decreased. Many CLAs and CSLAs have separate designs that have been discussed in the literature. A direct comparison of their results based on the design metrics would be helpful. To enable a comparison, we constructed 32-bit accurate and approximate additions in homogeneous and hybrid CLAs, as well as CSLAs with and without the BEC converters. We looked at a 32/28 nm complementary metal-oxide semiconductor (CMOS) process with a typical-case process-voltage-temperature (PVT) specification for the gate-level implementations. The findings indicate that, in terms of speed and power, the hybrid CLA/RCA architecture is preferable to the CLA and CSLA structures for performing precise and approximative additions.

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

I. INTRODUCTION

Addition is ubiquitous in microprocessing and digital signal processing hardware. It is performed using an adder. In real applications, adders are fast and low current. In this context, carry-ahead adders and carry-select adders are two common fast adders. Low power adder architecture [1]. Two variants of the carry-ahead adder (CLA) are common. recursive CLA (RCLA) [2] and block CLA (BCLA) [3]. These speed performance The CLA architecture can be improved by adopting a compact hybrid CLA architecture. Ripple Carry Adder (RCA) in least significant adder bit position as one or more alternatives Sub-CLA [4]. Furthermore, the improvement in speed performance is Reduced power loss in hybrid CLA. Therefore RCLA/RCA and BCLA/RCA is also called high-speed low-power hybrid CLA architecture [5].

Two variants of the carry-select adder (CSLA) [6] are common - the complete architecture Adder and 2:1 Other architectures with multiplexers (MUXes) and full adders 2:1 multiplexer, Binary to excess one code (BEC) converters [7,8]. In the literature, the design is uniformWe discuss hybrid CLA at gate level and transistor level. Similarly CSLA designI will explain when to use the BEC converter and when not to use it. However, the designs are different CLA and CSLA. 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.

In this communication, the physical realization of homogeneous and hybrid CLA and CSLA with and without BEC converters by considering matching 32-bit complement examples Accurate and approximate calculations. To perform approximate addition, lower OR Consider the proximity adder (LOA) from Ref. [10]. Various CLAs and CSLAs Physically implemented using 32/28 nm complementary bulk metal oxide semiconductor (CMOS) Process [11] is for typical process voltage-temperature (PVT) specifications with power supply A voltage of 1.05 V and an operating temperature of 25 °C to run the simulation.

The rest of this communication is divided into four sections. Section 2 describes CSLA architecture. Section 3 describes homogeneous and hybrid CLA architectures. Presenting Section 4 The simulation results obtained for CLA and CSLA are exact and in close agreement. addition. This is followed by the conclusions of Section 5.

II. CSLA ARCHITECTURES

The CSLA architecture divides the initial input bits into groups of equal or different size and divides the entire addition into many sub-additions that can be executed in parallel. When the initial and addend input bits of the CSLA are divided into groups of the same size, this is called "uniform CSLA", and when the initial and addend input bits of the CSLA are divided into groups of different sizes, this is It's called "non-uniform". -Uniform CSLA". References [7,8] suggest that a non-uniform CSLA is desirable for high speed and low power consumption.

Two types of CSLA architecture are common. The first uses dual RCAs with the appropriate size determined by the input partition, with RCAs in the least significant adder bit positions. Has a fixed carry-in from 1. Dual RCA outputs are routed to 2. One MUX with carry-out to service the previous input partition As the selected input of the MUX belonging to the current input partition - this architecture This means that no BEC converter is used. In the second architecture, One RCA in the least significant bit position of the adder using the specified number of appropriately sized RCAs Occupied by the input partition with a fixed carry-in of 0. The outputs of these RCAs are shown. To BEC converter [7]. This increments the output of the RCA by 1 in binary. This architecture is This demonstrates the use of the BEC converter. The selection of the output of the RCA with fixed carry-in 0 or the output of the BEC converter to generate the desired sum in CSLA_BEC is performed using a MUX. For MUXes associated with input partitions, the carry-out from the previous input partition serves as the common select input.

A block diagram of an optimal non-uniform 32-bit CSLA is shown in Figure 1a. A gate-level implementation of the one full adder and two 4-bit CSLA example includes: One MUX (CSLA_NOBEC) and another MUX with full adder, 2:1 MUX and BEC converter (CSLA_BEC) are shown in Figure 1b,c respectively. Details of the internal gate levels of the example 5-bit BEC converter are also shown within the green dotted rectangle in Figure 1c.

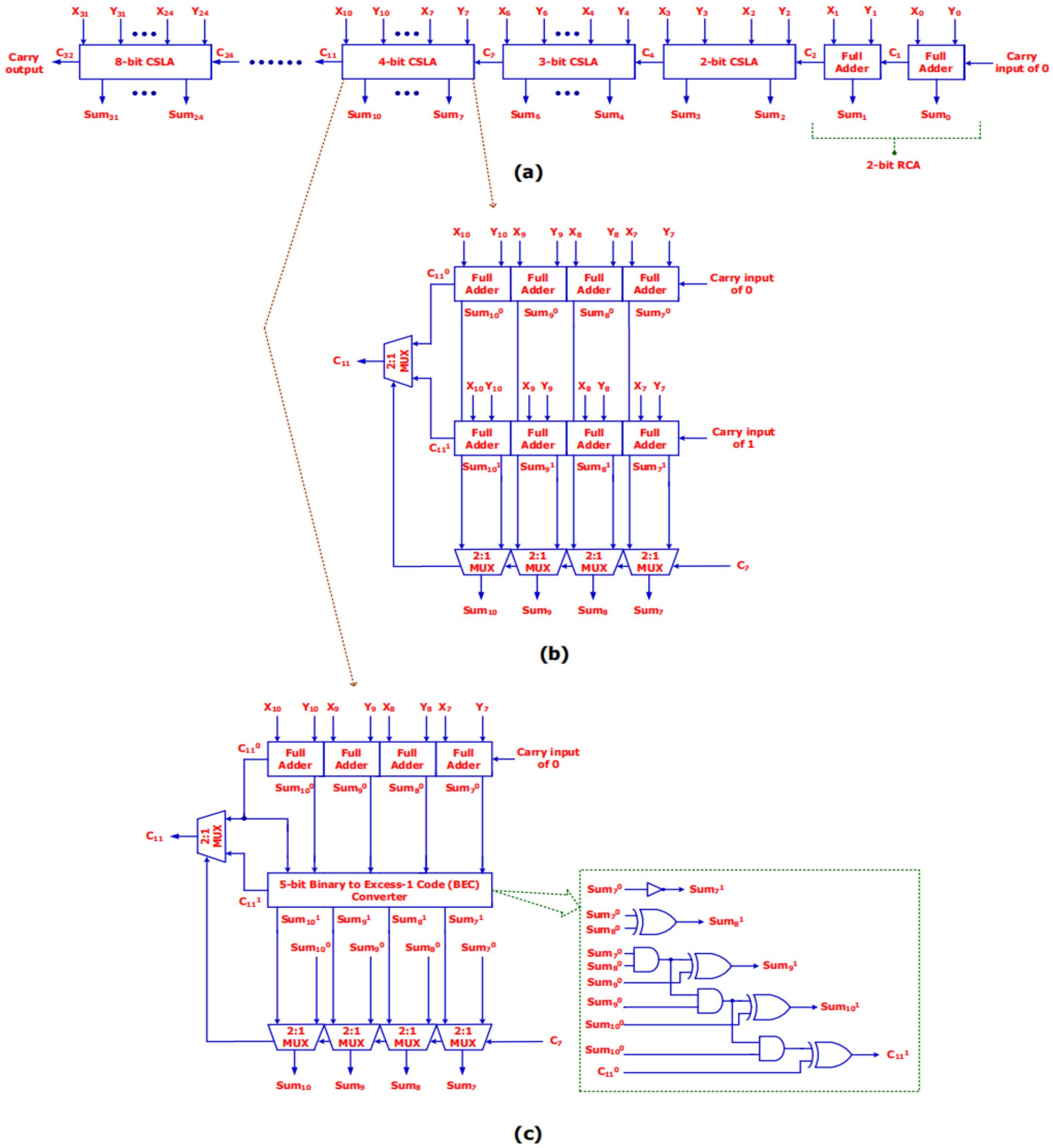


Figure 1. (a) Non-uniform 32-bit carry-select adder (CSLA) based on an optimum 8-7-6-4-3-2-2 input partition; (b) internal detail of a 4-bit (sub-)CSLA comprising full adders and 2:1 multiplexers (MUXes); (c) internal detail of a 4-bit (sub-)CSLA comprising full adders, 2:1 MUXes, and a 5-bit binary-to-excess-1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

III. Homogeneous CLA and Hybrid CLA/RCA Architecture

3.1 RCLA and RCLA/RCA Architectures

The recursive carry-lookahead equations are the foundation of the RCLA architecture [1-3]. Equations (1) to (4) provide the generalised logic formulations of the propagate and generate functions, the lookahead carry output, and the sum output (4). Equations (1) through (4) denote an arbitrary adder bit position by I, adder input bits by X and Y, propagate and generate functions by P and G, carry signal by C, and sum output by SUM. The associated augend and addend input bits are subjected to an exclusive-OR (XOR) operation to obtain the propagate function. The accompanying augend and addend input bits are logically combined to form the generate function. The sub-CLA-corresponding lookahead carry output (CI+1) is recorr using the propagate and generate functions, the carry input, and Equation (3) (CI).

The associated propagation function and the carry input bit are XORed to produce the sum output bit.

$$PI = XI \oplus YI \qquad (1)$$

$$GI = XIYI \qquad (2)$$

$$CI+1 = GI + PIGI-1 + ... + PIPI-1 ... P0CI \qquad (3)$$

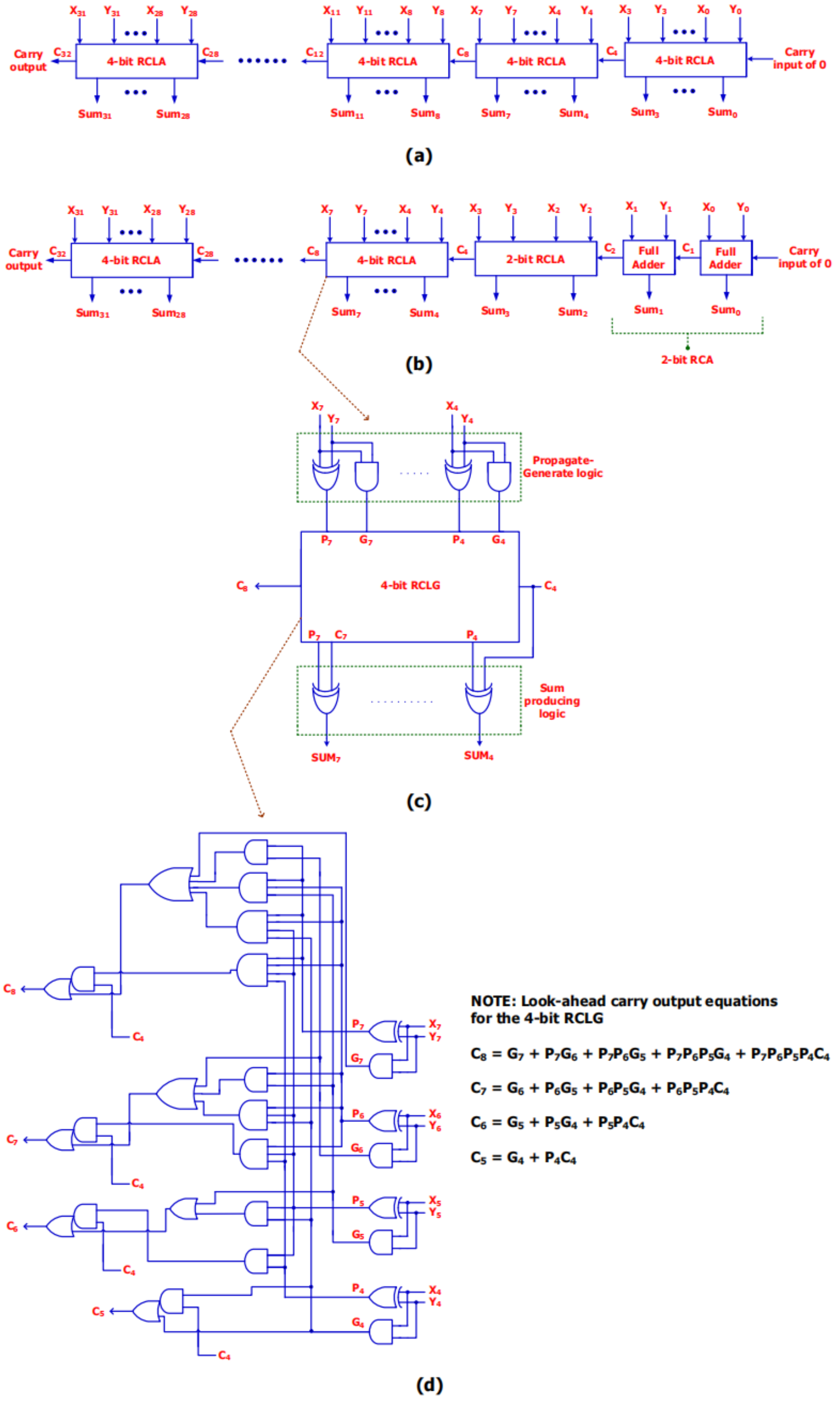
$$SUMI = PI \oplus CI \qquad (4)$$

Equation (3) is essentially recursive since it may be used to determine the lookahead carry output for any bit location given knowledge of the carry input. The propagate-generate logic, which is represented by Equations (1) and (2), the recursive carry-lookahead generator (RCLG), which is represented by Equation (3), and the sum-producing logic, which is represented by Equation (4) make up an RCLA (4). An RCLA is typically built by cascading numerous small-sized (sub-)RCLAs. For instance, cascading eight 4-bit (sub-)RCLAs can result in a 32-bit RCLA. The physical realisation of a high-speed and low-power RCLA is of interest [5,] as are other realisations of the RCLA that are achievable, as mentioned in the references [4,5].

Figure 2a depicts a homogeneous 32-bit RCLA made up of eight 4-bit RCLAs with delay optimisations [5]. Figure 2b depicts the ideal may be noticed from Figure 2d that the maximum data path delay (also called the critical path delay) is encountered in producing C8, 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 fullhybrid 32-bit RCLA, which includes a 2-bit RCLA and two complete adders in the least significant bit positions.

Figure 2c shows the internal components of a sample 4-bit RCLA, which include the propagate-generate logic, a 4-bit RCLG, and the sum-producing logic. Figure 2d [5] depicts the gate-level realisation of a 4-bit delay-optimized RCLG. As seen in Figure 2d, four lookahead carry output signals—C5, C6, C7, and C8—are produced based on the carry input C4 and in accordance with Equation (3). C4, C5, C6, and C7 of these are XORed with the equivalent propagate the P4, P5, P6, and P7 functions to generate the corresponding sum output bits, or SUM4 to SUM7. The only lookahead carry output signal, C8, is sent to the following 4-bit sub-RCLA, where it is used as the carry input. M propagate and create functions, M lookahead carry signals, and (M-1) lookahead carry signals are all produced in an M-bit (sub-)RCLA, which is then used internally to produce the M total output bits. The subsequent sub-RCLA will only get the most important lookahead carry signal as its carry input.

Figure 2d shows that the production of C8, which is determined by the total 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, encounters the maximum data path delay, also known as the critical path delay. This critical path would be encountered by the 4-bit (sub-)RCLA that makes up the least significant N-bit RCLA.delay. Yet, the propagation delay of just one AO21 complex gate would be the smallest data channel delay that the next 4-bit (sub-)RCLAs would experience. Hence, it might be advantageous to substitute a reduced (sub-)RCLA and any full adders for the least important M-bit (sub-)RCLA in an N-bit RCLA. As a result, Figure 2b depicts the substitution of a 4-bit (sub-)RCLA with a 2-bit (sub-)RCLA and two full creating a hybrid RCLA/RCA architecture using adders. The homogeneous RCLA's critical path delay, silicon area, and average power dissipation could all be decreased using the hybrid RCLA/RCA architecture [5].


(a)
Carry output
4-bit RCLA
Carry input of 0
X_{31} Y_{31} X_{28} Y_{28}
C_{32} C_{28} C_{12} C_8 C_4
Sum_{31} Sum_{28} Sum_{11} Sum_8 Sum_7 Sum_4 Sum_3 Sum_0
(b)
2-bit RCLA
Full Adder
2-bit RCA
C_2 C_1
(c)
Propagate-Generate logic
4-bit RCLG
Sum producing logic
P_7 G_7 P_4 G_4 C_7
SUM_7 SUM_4
(d)
C_8 C_7 C_6 C_5 C_4

NOTE: Look-ahead carry output equations for the 4-bit RCLG

$$C_8 = G_7 + P_7G_6 + P_7P_6G_5 + P_7P_6P_5G_4 + P_7P_6P_5P_4C_4$$

$$C_7 = G_6 + P_6G_5 + P_6P_5G_4 + P_6P_5P_4C_4$$

$$C_6 = G_5 + P_5G_4 + P_5P_4C_4$$

$$C_5 = G_4 + P_4C_4$$

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 lookahead carry output equations.

3.2 BCLA and BCLA/RCA Architectures

Another sort of CLA that uses the recursive carry-lookahead equation for synthesis is the BCLA [3], also known as the section-carry-based carry-lookahead adder (SCBCLA) [4,5]. An N-bit BCLA is built using numerous little M-bit (sub-)BCLAs, just like the RCLA. Figure 3a depicts a homogeneous 32-bit BCLA built from eight 4-bit (sub-)BCLAs that have been delay-optimized, whereas Figure 3b depicts a hybrid 32-bit BCLA/RCA.

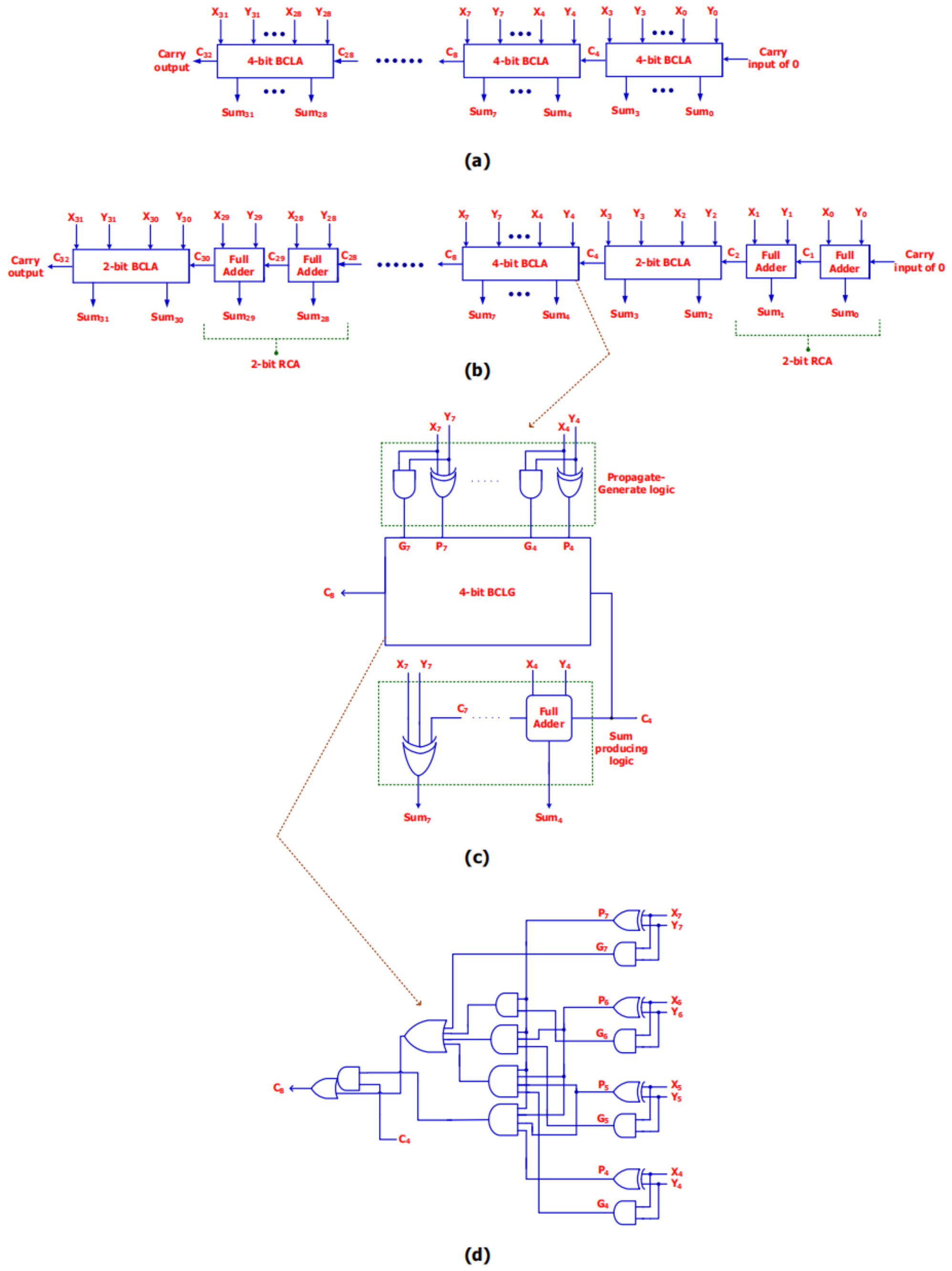


Figure 3. (**a**) Homogeneous 32-bit block carry-lookahead adder (BCLA); (**b**) optimum hybrid 32-bit BCLA/RCA; (**c**) internal detail of a 4-bit (sub-)BCLA; (**d**) gate-level detail of a delay-optimized 4-bit block carry-lookahead generator (BCLG), which is custom-realized.

The BCLA, however, is distinct from the RCLA. A lookahead carry input from a previous (sub) BCLA is received by an M-bit BCLA (also known as the sub-BCLA), which then generates one lookahead carry output for the following

DOI: 10.17148/IJARCCE.2023.12449

(sub-)BCLA. Note that an M-bit RCLA, on the other hand, generates M lookahead carry outputs. If M = 4, Figure 3c displays an illustration of an M-bit (sub-)BCLA.

As seen in Figure 3c, an M-bit BCLA is made up of the propagate-generate logic, an M-bit block carry-lookahead generator (BCLG), and the sum-producing logic. One lookahead carry output is created using the M-bit (sub-)BCLA's carry input. To generate the necessary sum output bits, the carry input is simultaneously processed by a cascade of (M-3) full adders and a three-input XOR gate that resembles a sub-RCA. Figure 3d displays the gate level information for a 4-bit BCLG with delay optimisation. Figure 3d is the consequence of discarding the remainder of the lookahead carry output logic and extracting the logic that corresponds to C8 from Figure 2d. Similar to what is shown in Figure 2d, the logic expression for C8, which corresponds to Figure 3d, is the same.When N modulo 4 4-bit (sub-)BCLAs are cascaded, a homogeneous N-bit BCLA is created.

Since 4 is equal to 0, the critical path in the least significant 4-bit (sub-)BCLA would correspond to the output of the lookahead carry and consist of a final AO21 complex gate, a final XOR gate, a final AND gate, and a final OR gate. One solution would be to swap out the least significant 4 bit (sub-)BCLA and any full adders for smaller (sub-)BCLAs in order to shorten the critical path latency. The resulting ideal hybrid BCLA/RCA design for 32-bit addition is shown in Figure 2b. Figure 2b shows that, similar to what was done for the least significant 4-bit (sub-)BCLA, the most significant 4-bit (sub-)BCLA is similarly substituted with a 2-bit (sub-)BCLA and two complete adders. This is due to the fact that the most A large 4-bit (sub-)BCLA would consist of one three-input XOR gate and three complete adders. In contrast, the critical route in Figure 2b would come into contact with one AO21 gate and two complete adders, which helps to somewhat shorten the critical path latency. Similar to the hybrid RCLA/RCA design, the hybrid BCLA/RCA architecture would aid in decreasing the homogeneous BCLA architecture's average power dissipation, silicon area, and critical path time.

IV. RESULTS AND DISCUSSION

To compare the performances of the homogeneous and hybrid 32-bit CLAs and CSLAs in terms of design metrics, a semi-custom ASIC-style standard cell-based physical implementation was taken into consideration. A 32/28 nm CMOS technique was used to implement all of the adders [11]. The CLAs and CSLAs were realised using the complete adder and 2:1 MUX available in the digital cell library [11]. 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. The critical path delay, silicon area, and average power dissipation of the adders were estimated. The average power dissipation was calculated using 1000 identical random input vectors.Using the same test bench, the adders were tested at 200 MHz time steps every 5 ns. The average power dissipation was then calculated using the switching activity recorded during the functional simulations. Also estimated were the essential path delays and area occupancies. When estimating the design metrics, the default wire load, or the maximum wire load selection group “predcaps,” was automatically taken into account. The designs were implemented and simulated using Synopsys Electronic Design Automation (EDA) tools Design Vision and VCS, and the design metrics were estimated using PrimeTime. The average power dissipation was precisely calculated using PrimeTime’s time-based power analysis mode.

A digital circuit or system's low-power design efficiency can be measured using the power-delay product (PDP) and energy-delay product (EDP) [12]. The PDP and the EDP of the adders were determined and normalised in light of this. In order to normalise the data, the maximum computed PDP and EDP values were taken into account as the baseline values, and these values were then utilised to divide the actual PDP and EDP values of all the adders. Consequently, the PDP and EDP parameters’ least fractional value.

4.1 Results for Accurate Addition

Using the gates of the 32/28 nm standard digital cell library, precise 32-bit CSLAs for the CSLA NOBEC and CSLA BEC architectures outlined in Part 2 were physically implemented [11]. Hybrid RCLA/RCA and BCLA/RCA, as well as precise 32-bit homogeneous RCLA and BCLA, are The same digital cell library was used to physically materialise the concepts outlined in Section 3 as well. The style Table 1 provides estimated metrics for the precise 32-bit adders. PDP and EDP normalised The adders’ plots are displayed in Figures 4a and 4b, respectively. Figure 4’s red bars highlight the best. This corresponds to the hybrid CLA among the CSLAs and CLAs selected for physical implementation RCLA/RCA construction.

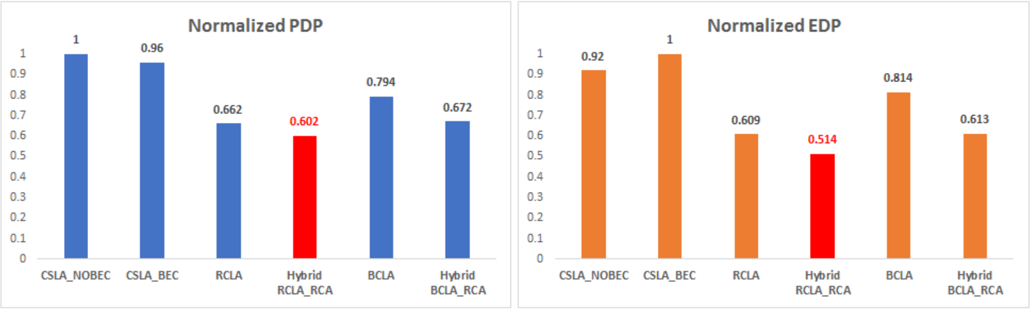
According to Table 1, the hybrid RCLA/RCA design is superior to the others in terms of delay and power, and as a result, it has the lowest values of PDP and EDP when compared to the other CLAs and CSLAs. The hybrid 32-bit RCLA/RCA reports a 15.6% reduction and a 9.1% decrease in PDP.

Compared to its nearest analogue, the homogenous 32-bit RCLA, in the EDP. Moreover, Compared to the latter, the former requires 6% less silicon. It was observed that the 32-bit BCLA required less space.compared to the 32-bit RCLA. Due to the 4-bit BCLA in Figure 3c's need for 22.6% less silicon.in Figure 2c for the practical manifestation than the 4-bit RCLA. Regarding area occupancy,We discovered that CSLA NOBECbest of the CLAs and CSLAs taken into account. Despite this, the hybrid RCLA/PDP RCA's and EDP are considerably less than the comparable values of By 39.8% and 44.1%, respectively, above the CSLA NOBEC.Table 1 shows the design metrics for precise 32-bit addition. The carry-select adder CSLA NOBEC Without a binary-to-excess-1 code converter, and CSLA BEC, a carry-select adder with a binary-to-excess-1 code converter. Code converter; RCA (ripple-carry adder); RCLA (recursive carry-lookahead adder); BCLA (block carry adder);Adder with carry-lookahead.Area (m2) and Adder Delay Type (ns) (Power [W]) 61.51 CSLA NOBEC 1.13 418.32 1.28 CSLA BEC 459.49 52.12 RCLA 1.13 646.54 40.70 RCLA/RCA hybrid 1.05 607.91 39.82 BCLA 1.26 500.16 43.80 BCLA/RCA hybrid 1.12 457.97 41.72 2018, 7, x FOR PEER REVIEW 8 of 12 Electronics CSLA NOBEC and CSLA BEC architectures' precise 32-bit CSLAs are covered inPhysical realisations of Section 2 were made utilising the gates from the conventional digital cell library at 32/28 nm [11].

Hybrid RCLA/RCA and BCLA/RCA, as well as precise 32-bit homogeneous RCLA and BCLA, are The same digital cell library was used to physically materialise the concepts outlined in Section 3 as well. The style Table 1 provides estimated metrics for the precise 32-bit adders. PDP and EDP normalisedThe adders' plots are displayed in Figures 4a and 4b, respectively. Figure 4's red bars highlight the best.among the CLAs and CSLAs taken into consideration for physical execution, which RCLA/RCA hybrid architecture.The hybrid RCLA/RCA architecture in Table 1 is preferred over the RCLA/RCA architecture in terms of delay and power.This allows it to have the lowest PDP and EDP values compared to the rest, several CLAs plus CSLAs. The hybrid 32-bit RCLA/RCA reports a 15.6% reduction and a 9.1% decrease in PDP.compared to its nearest analogue, the homogenous 32-bit RCLA, in the EDP. Moreover,Compared to the latter, the former requires 6% less silicon. It was observed that the 32-bit BCLA required less space.compared to the 32-bit RCLA. Due to the 4-bit BCLA in Figure 3c's need for 22.6% less silicon.in Figure 2c for the practical manifestation than the 4-bit RCLA. Regarding area occupancy,The finest CLA and CSLA that was taken into consideration was CSLA NOBEC. Despite this, The hybrid RCLA/PDP RCA's and EDP are much lower than the respective parameters.by 39.8% and 44.1%, respectively, of the CSLA NOBEC.

Table 1. Design parameters for precise 32-bit addition. Carry-select adders with binary-to excess-1 code converters are designated as CSLA NOBEC, CSLA BEC, RCLA, RCA, and BCLA, respectively. Recursive carry-lookahead adders, ripple-carry adders, and block carry-lookahead adders are also available.

Type of Adder	Delay (ns)	Area (μm^2)	Power (μW)
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



(a) (b)

Figure 4. Normalized (**a**) power–delay product (PDP) plots, and (**b**) energy–delay product

DOI: 10.17148/IJARCCE.2023.12449

4.2 Results for Approximate Addition

The LOA given in [10] was used to do approximate addition, as its effectiveness was confirmed using neural network and fuzzy applications. Moreover, it was discovered that the LOA provided the best cost-error trade-off in the stochastic regime [13].

The LOA divides the input bits into two parts, the most significant accurate adder part and the least significant approximate adder part, for processing. Here, for instance, we took into account an equal bi-partition of the input bits to actualize the LOA; 16 bits were assigned to the exact adder component and 16 bits to the approximation adder part. A succession of two-input OR gates make up the approximation adder component, and each one executes a logical disjunction of the matching augend. addend input bits, etc.

The AND operation is applied on the most significant bit pair of the approximate adder part, and the result is provided as the carry input for the accurate adder part. Any high-speed adder can be used to realise the accurate adder component. The CLA and CSLA architectures covered in the previous parts were used to implement the accurate adder component in this communication. In Figure 5a-e, the resulting LOA structures are displayed.

The LOA, which has a 16-bit non-uniform CSLA for the correct adder component, is shown in Figure 5a. To realise the non-uniform CSLA for the accurate adder component, [8] an ideal 5-4-3-2 2 input partition was taken into consideration. Either the CSLA NOBEC or the CSLA BEC architecture can be used with the 16-bit CSLA. The accurate adder part's homogeneous RCLA, which consists of four 4-bit (sub-)RCLAs, is shown in Figure 5b. Figure 5c illustrates the use of the hybrid RCLA/RCA for the accurate adder part. In the least significant nibble position of the accurate adder part, a 2-bit (sub-)RCLA and two full adders are used, and three 4-bit (sub-)RCLAs are used for the more significant bit 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. Table 2 shows that the LOA with the hybrid RCLA/RCA in the accurate adder component outperformed the others in terms of optimisations for delay and power.

As a result, when compared to LOAs that used different CLAs or CSLAs for the accurate adder component, the hybrid RCLA/RCA LOA recorded the lowest values of PDP and EDP. In comparison to its closest counterpart, the 32-bit LOA incorporating a homogeneous RCLA for the accurate adder component, the hybrid RCLA/RCA for the 32-bit LOA reported 11.3% and 17.1% reductions in the PDP and EDP, respectively. Also, the former's silicon footprint was 10.8% smaller than the latter's.

DOI: 10.17148/IJARCCE.2023.12449

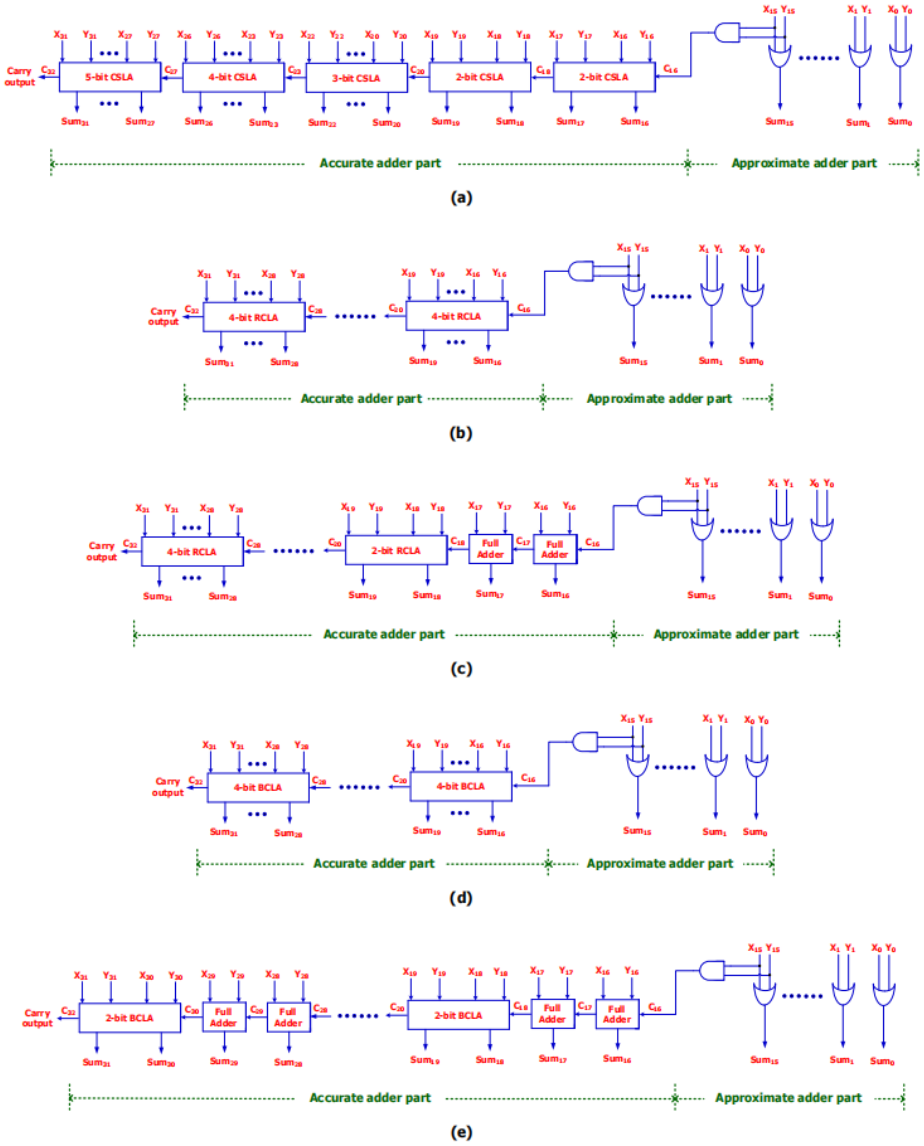


Figure 5. Different 32-bit lower-part-OR approximate adders (LOAs) consisting of diverse 16-bit accurate adder parts and a common 16-bit approximate adder part.

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 (μm^2)	Power (μW)
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

DOI: 10.17148/IJARCCE.2023.12449

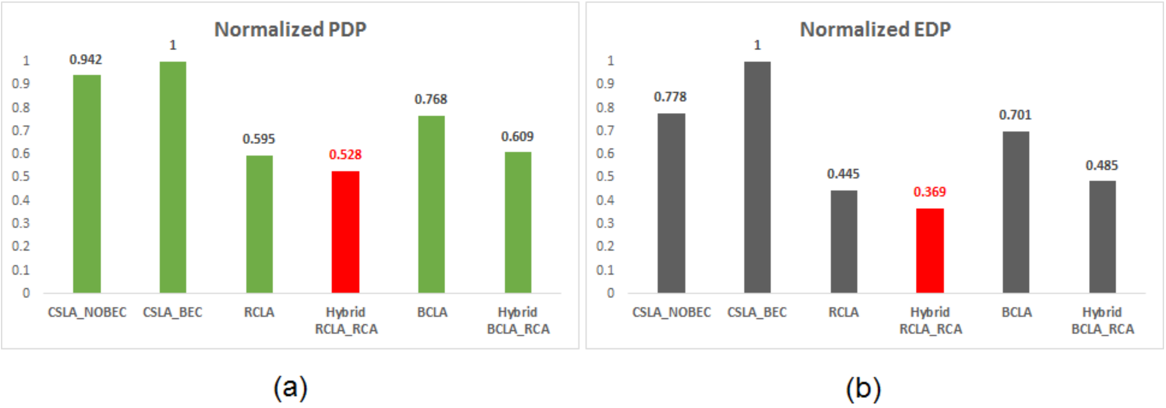


Figure 6. Normalized (a) PDP plots, and (b) EDP plots of approximate 32-bit adders (LOAs)—the types of adders used for the 16-bit accurate adder part of LOAs are mentioned on the X-axis.

The 32-bit LOA using the hybrid BCLA/RCA design for the accurate adder component was the best of the bunch in terms of silicon area, taking up 6.2% less space than its nearest competitor, the 32-bit LOA using the CSLA NOBEC architecture. This is mostly due to the fact that, in the case of the former, only two 4-bit (sub-)BCLAs were utilised for the two intermediate nibble places of the accurate adder section, and that the remaining bit positions were filled by two 2-bit (sub-)BCLAs and four complete adders. The combination of a 2-bit (sub-)BCLA and two complete adders takes up 33.7% less space than a 4-bit (sub-)BCLA. This results in a less area requirement for the accurate adder's 32-bit LOA, which consists of the 16-bit BCLA/RCA. CSLA NOBEC is a 16-bit register used for the accurate adder component in the 32-bit LOA as opposed to this. However, the PDP and EDP values were significantly reduced by the LOA with a 16-bit RCLA/RCA for the correct adder component.ln comparison to the comparable design characteristics of the LOA employing a 16-bit BCLA/RCA for the accurate adder component, by 13.3% and 23.9%, respectively.

IV. CONCLUSIONS

In this communication, the implementation of high-speed, low-power CLAs and CSLAs was reviewed, and the performances of the two were contrasted by taking into account precise and approximative 32-bit additions. The comparisons reveal that, in terms of latency and power dissipation, the hybrid RCLA/RCA architecture outperforms the various CLA and CSLA systems. Another benefit of the CLA design is that, according to Reference [9], the CLA requires a significantly smaller number of input patterns than the CSLA when evaluating stuck-at faults. The utility of the hybrid RCLA/RCA design can also be thoroughly investigated by taking into account a variety of digital signal processing processes, many of which frequently involve adds and multiplications. The family of high-speed parallel-prefix adders is also an outgrowth of this brief communication.

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, USA, 2003;2. 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]. ASIC-based implementation of synchronous section-carry based carry lookahead adders. Balasubramanian, P.; Mastorakis, N.E. Mladenov, V., ed., Recent Developments in Circuits, Systems, Signal Processing, and Communications; WSEAS Press: Athens, Greece, 2016; pp. 58-64; ISBN 978-1618043665.
[5]. Design of better synchronous section-carry based carry lookahead adders, Balasubramanian,P.metric for merit. 155–164 in WSEAS Trans. Circuits Syst. 2016.
[6]. Carry-select adder by Bedrij, O.J. EC-11, 340–346 in Trans. Electron. Comput 1962.
[7]. Kittur, H.M. and Ramkumar, B. choose adder with low power and small area. Extremely Huge Scale in IEEE Trans.2012, 20, 371-375, Integr. (VLSI) Syst.
[8]. Area-delay-power efficient carry select adder by Mohanty, B.K. and Patel, S.K. IEEE Transactions on Circuit Systems II 2014, Exp. Briefs 61, 418-422.
[9]. Implementation and test pattern by Saini, V.K., Akhter, and Chauhan.