

# Factorized Carry Lookahead Adders

P. Balasubramanian, D. L. Maskell

School of Computer Science and Engineering, Nanyang Technological University, Singapore 639798  
{balasubramanian, asdouglas}@ntu.edu.sg

**Abstract**—New factorized carry lookahead adders corresponding to the regular carry lookahead adder (RCLA) architecture viz. the factorized regular carry lookahead adder (FRCLA), and the block carry lookahead adder (BCLA) architecture viz. the factorized block carry lookahead adder (FBCLA) are presented. The idea behind the proposed factorized carry lookahead adders is discussed and example implementations are provided. The RCLA, BCLA, FRCLA and FBCLA were realized using the gates of a 32/28nm CMOS standard digital cell library. The results show that the proposed FRCLA achieves an average reduction in the power-delay product i.e., energy by 7.85% for 32- and 64-bit additions compared to the best among the rest.

## I. INTRODUCTION

Addition forms the basis of computer arithmetic, and addition is a fundamental operation encountered in micro-processing and digital signal processing units. Addition is accomplished using an adder, and the carry lookahead adder (CLA) is an important member of the family of high-speed adders [1].

Different implementations of the CLA at the transistor and gate levels have been discussed in the literature [2–9], based on static and dynamic MOS-based implementation styles. Designs of CLAs using post-CMOS technologies such as quantum dot cellular automata, memristor, optical, carbon nanotube, and vertically-stacked nanowire transistors have also been described [10–15]. Further, the designs of CLAs based on different architectural styles have been described [16, 17], and the RCLA and BCLA architectures are prominent among them. The RCLA is preferable for high speed and low power while the BCLA is preferable for reduced area.

In this paper, we propose new factorized carry lookahead adders, i.e., FRCLA based on the RCLA architecture, and FBCLA based on the BCLA architecture. The FRCLA and the FBCLA are based on recursive carry lookahead equations, which are subject to algebraic factorization prior to physical synthesis. Logic factorization, prior to synthesis, helps to optimize the lookahead carry logic better and gives rise to a generic and regular CLA topology, which is likely to be advantageous for physical synthesis using any implementation technology, be it CMOS or post-CMOS. We show how the proposed FRCLA is beneficial than the other CLA architectures by considering 32-bit and 64-bit additions as examples. We utilized the Synopsys 32/28nm CMOS standard digital cell library [18] for physically implementing the various CLAs.

In the remainder of this paper, Section II discusses the RCLA topology and Section III discusses the BCLA topology, in brief. The proposed FRCLA and FBCLA are discussed in Section IV. The design metrics estimated for the CLAs corresponding to 32-

bit and 64-bit additions are given in Section V, and finally we conclude in Section VI.

## II. RCLA ARCHITECTURE

In general, an N-bit RCLA [17] is constructed by cascading N/M M-bit (sub-) RCLAs, where N and M are even, and N modulo M equals 0. Assuming M to be 4, Fig. 1a portrays the architecture of a N-bit RCLA. The most significant lookahead carry output produced by a M-bit RCLA is given as the carry input for the successive M-bit RCLA. In a M-bit RCLA, M lookahead carry outputs are produced and the (M–1) lookahead carry outputs are XOR-ed with the corresponding propagate signals to produce the respective sum output bits.

Fig. 1c shows the gate-level realization of a delay-optimized 4-bit RCLA [19], involving simple and complex gates of the standard digital cell library [18]. Based on physical realization using the gates of the digital cell library [18], Fig. 1c requires  $80.82\mu\text{m}^2$  of silicon. In Figs. 1 and 2,  $P_0$  to  $P_3$  represent the propagate signals,  $G_0$  to  $G_3$  represent the generate signals, and  $C_0$  represents the carry input.  $C_1$  to  $C_4$  represent the lookahead carry outputs, with  $C_4$  being the lookahead carry input for the successive 4-bit RCLA/BCLA/FRCLA/FBCLA.  $SUM_0$  to  $SUM_3$  represent the corresponding sum output bits in Figs. 1 and 2. The propagate, generate, and lookahead carry output signals are governed by the following equations. Among these, (3) is recursive by nature i.e., any lookahead carry output bit can be inferred based on (3). In the equations, sum refers to logical disjunction (i.e., OR) and product refers to logical conjunction (i.e., AND). The symbol  $\oplus$  signifies exclusive-OR (i.e., XOR).

$$P_K = A_K \oplus B_K \quad (1)$$

$$G_K = A_K B_K \quad (2)$$

$$C_{K+1} = G_K + P_K G_{K-1} + \dots + P_K P_{K-1} \dots C_0 \quad (3)$$

$$SUM_K = P_K \oplus C_K \quad (4)$$

The 4-bit RCLA shown in Fig. 1c, which represents the least significant 4-bit RCLA of Fig. 1a, will encounter a propagation delay that is equal to the sum of the propagation delays of a 2-input XOR gate, a 4-input AND gate, a 4-input OR gate, and an AO21 complex gate. Referring to Fig. 1a, the intermediate 4-bit RCLAs, excepting the most significant 4-bit RCLA, will encounter the propagation delay of just the AO21 gate. The last 4-bit RCLA in Fig. 1a will however encounter the propagation delay of an AO21 complex gate and a 2-input XOR gate.

This work is supported by the Ministry of Education (MOE), Singapore under grants MOE2017-T2-1-002 and MOE2018-T2-2-024.

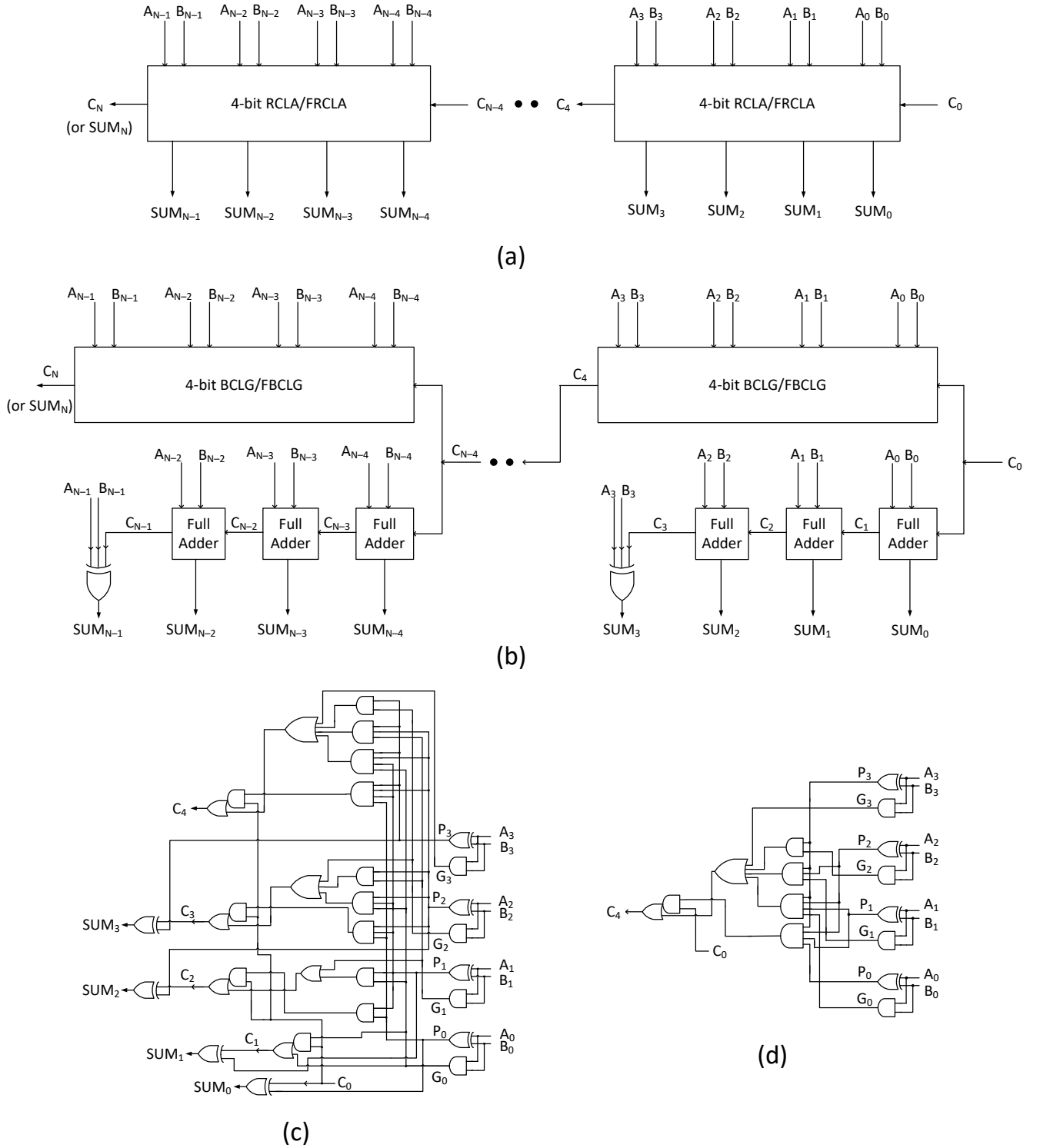


Figure 1. Block schematics of: (a) N-bit RCLA/FRCLA constructed using  $(N/4)$  4-bit RCLAs/FRCLAs where  $N$  is even and  $N \bmod 4 = 0$ , and (b) N-bit BCLA/FBCLA, constructed using  $N/4$  4-bit BCLAs/FBCLAs, where  $N \bmod 4 = 0$ . Gate-level realizations of: (c) 4-bit RCLA, and (d) 4-bit BCLG. The 4-bit BCLG shown in (d) is used in (b) to realize the 4-bit BCLA, which is subsequently duplicated to realize an N-bit BCLA. The full adder and 3-input XOR gates of [18] are also used to realize the BCLA/FBCLA shown in (b). The AO21 gate implements  $V = XY + Z$ ;  $X$ ,  $Y$  and  $Z$  are the inputs and  $V$  is the output.

### III. BCLA ARCHITECTURE

The BCLA architecture [16] shares some similarity with the RCLA architecture but there are some differences. To shed the

light on these, we make a comparison between Fig. 1b, which shows a N-bit BCLA, and Fig. 1a, which shows a N-bit RCLA. In Fig. 1b, a N-bit BCLA is constructed using  $N/4$  4-bit BCLAs where  $N$  modulo 4 equates to 0. In Fig. 1b, the lookahead carry

output produced by a 4-bit block carry lookahead generator (BCLG) is supplied as the carry input for the successive 4-bit BCLA. A M-bit (i.e., sub-) BCLA comprises a M-bit BCLG and a M-bit carry-ripple type adder. The carry input to a M-bit BCLA is serially processed by (M-1) full adders and a 3-input XOR gate to produce the respective sum output bits. The full adder and 3-input XOR gate present in the standard cell library [18] are also used to construct the BCLA and the FBCLA.

The gate-level realization of a 4-bit BCLG is shown in Fig. 1d. The 4-bit BCLA requires  $62.52\mu\text{m}^2$  of silicon based on [18]. The logic corresponding to  $C_4$  of Fig. 1c is extracted and shown as the 4-bit BCLG in Fig. 1d. Unlike a M-bit RCLA where M lookahead carry outputs are produced, in a M-bit BCLA only one lookahead carry output is produced which is passed on to the successive M-bit BCLA as its carry input. In a (F)BCLA, the carries generated via lookahead propagate between blocks and carries within the blocks ripple to produce the requisite sum output bits. On the other hand, in a (F)RCLA, the carries ripple between sections, and the carries generated via lookahead within a section are XOR-ed with the corresponding propagate signals to produce the respective sum output bits.

#### IV. FACTORIZED CARRY LOOKAHEAD ADDERS

The proposed FRCLA is a derivative of the regular RCLA and the architecture of the former is like the latter, as shown in Fig. 1a. Therefore, equations (1), (2) and (4) given earlier hold well in the case of the FRCLA. However, unlike the RCLA, the lookahead carry output equation characterized by equation (3) is factorized prior to physical synthesis. To shed the light on this, we give the factorized lookahead carry output expressions corresponding to a 4-bit FRCLA below.

$$C_4 = [G_3 + P_3\{G_2 + P_2(G_1 + P_1G_0)\}] + (P_3P_2P_1P_0) C_0 \quad (5)$$

$$C_3 = \{G_2 + P_2(G_1 + P_1G_0)\} + (P_2P_1P_0) C_0 \quad (6)$$

$$C_2 = (G_1 + P_1G_0) + (P_1P_0) C_0 \quad (7)$$

$$C_1 = (G_0 + P_0C_0) \quad (8)$$

By extracting the logic kernels from the factorized equations, we have,  $W_1 = G_1 + P_1G_0$ ,  $W_2 = G_2 + P_2W_1$ ,  $W_3 = G_3 + P_3W_2$ ,  $X_1 = P_1P_0$ ,  $X_2 = P_2X_1$  and  $X_3 = P_3X_2$ . Subsequently, (5) to (8)

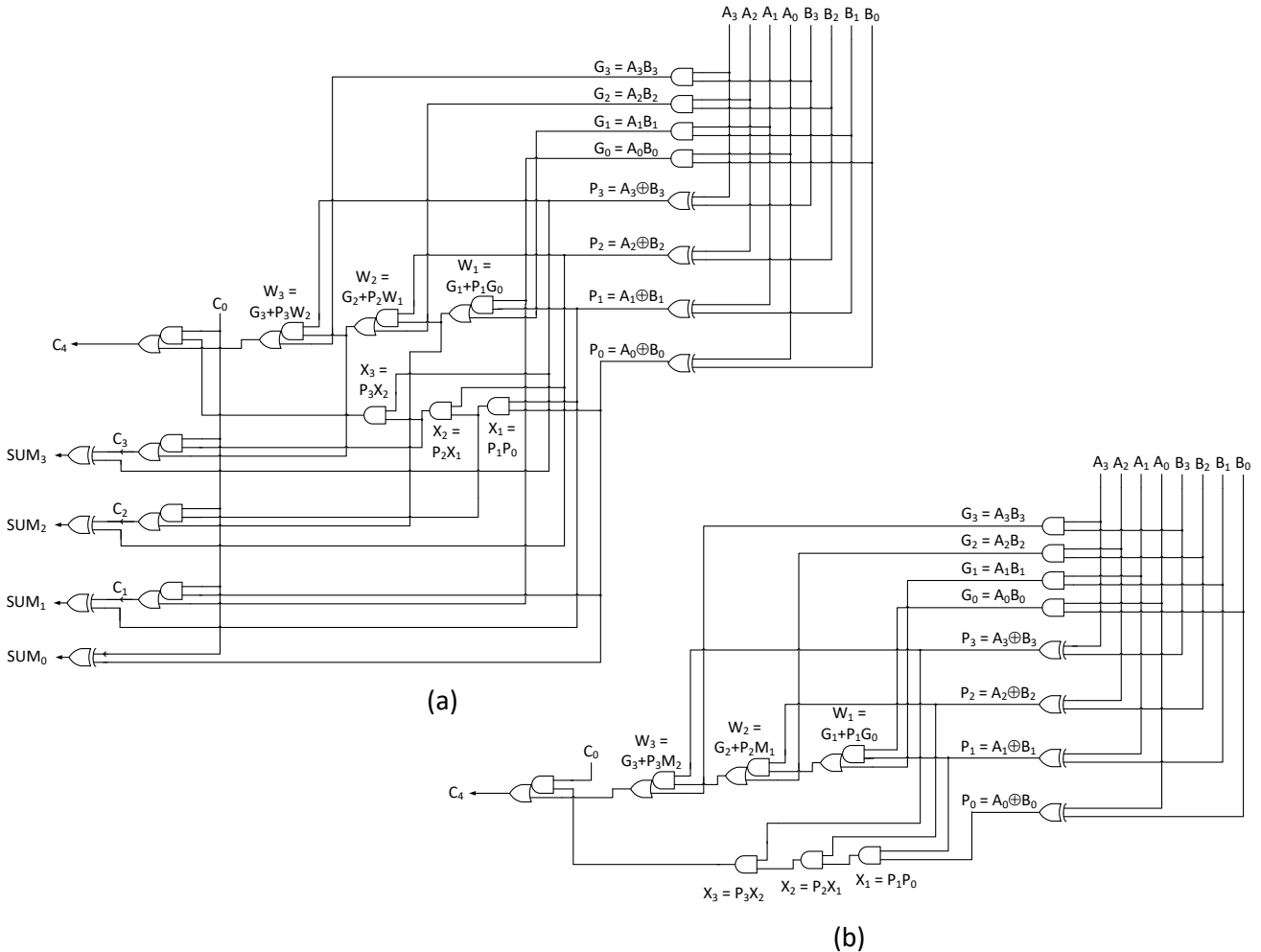


Figure 2. Gate-level realizations of proposed: (a) 4-bit FRCLA, and (b) 4-bit FBCLG. The logic corresponding to  $C_4$  is extracted from (a) and shown in (b). The 4-bit FRCLA is used to construct a N-bit FRCLA based on the architecture shown in Fig. 1a. The 4-bit FBCLA is realized using the 4-bit FBCLG, full adders and the 3-input XOR gate from the cell library. The 4-bit FBCLA is used to construct a N-bit FBCLA based on the architecture shown in Fig. 1b.

are transformed as follows:  $C_4 = W_3 + X_3C_0$ ;  $C_3 = W_2 + X_2C_0$ ;  $C_2 = W_1 + X_1C_0$ ; and  $C_1 = G_0 + P_0C_0$ . The transformed look ahead carry output equations indicate that all the lookahead carry outputs could be realized using a single AO21 gate in the final logic level, and only one AO21 gate will be encountered in any intermediate 4-bit FRCLA/FBCLA present in an N-bit FRCLA/FBCLA, thus facilitating high-speed carry propagation which has a positive impact on their speed performance. The gate-level realizations of the 4-bit FRCLA and the 4-bit FBCLA are shown in Figs. 2a and 2b. The 4-bit FRCLA and the 4-bit FBCLA require  $66.59\mu\text{m}^2$  and  $59.72\mu\text{m}^2$  of silicon respectively.

## V. IMPLEMENTATION RESULTS

32- and 64-bit RCLAs, BCLAs, FRCLAs and FBCLAs were implemented using the Synopsys 32/28nm bulk CMOS standard digital cell library [18]. About a thousand random input vectors corresponding to 32-bit and 64-bit additions were supplied to the adders at time intervals of 5ns (200MHz) through respective test benches to verify their functionalities. The switching activity data captured from the functional simulations were used to estimate the average power dissipation. The critical path delay and silicon area were also estimated, and the design metrics viz. power, delay, and area are given in Table I.

The power-delay product (PDP) is widely used to assess the low power/low energy attribute of a digital circuit/system. Since power and delay are desirable to be minimum, the PDP is also desired to be minimum. The PDP of the adders were calculated and normalized, given in Table I. To perform the normalization, the highest value of PDP corresponding to a N-bit CLA was considered as the baseline and this value was used to divide the actual PDP values of all the N-bit CLAs. Thus, the least value of PDP corresponding to a N-bit CLA indicates that it is the best among the rest of the N-bit CLAs.

TABLE I. DESIGN METRICS OF VARIOUS 32- AND 64-BIT CLAS

Type of CLA	Delay (ns)	Area ( $\mu\text{m}^2$ )	Power ( $\mu\text{W}$ )	Normalized PDP
<b>32-bit Adders</b>				
RCLA	1.13	646.54	40.70	0.79
BCLA	1.26	500.16	43.80	0.94
FRCLA	1.15	532.69	37.04	0.73
FBCLA	1.29	477.79	45.39	1
<b>64-bit Adders</b>				
RCLA	1.85	1293.08	81.68	0.86
BCLA	1.91	1000.31	87.60	0.95
FRCLA	1.88	1065.37	74.37	0.79
FBCLA	1.94	955.58	90.77	1

From Table I, it is noted that the FRCLA dissipates the least power for both 32- and 64-bit additions, and hence achieves the least PDP compared to the other CLAs thus conveying that it is more power- and energy-optimized than the rest. It is noted that the proposed FRCLA achieves reductions in the PDP by 7.6% for 32-bit addition and 8.1% for 64-bit addition compared to the best amongst the rest viz. RCLA. In terms of area, the FBCLA features the least silicon footprint among the rest of the CLAs

for 32- and 64-bit additions. This is mainly because compared to the 4-bit RCLA, the 4-bit BCLA and the 4-bit FRCLA, the 4-bit FBCLA requires less area by 26.1%, 4.5% and 10.3% respectively. In terms of the critical path delay, the FRCLA is neck and neck with the RCLA, while the FRCLA requires less area and dissipates less power than the RCLA. The enhanced optimizations in design parameters achieved by the FRCLA over the RCLA is attributed to the logic factorization done prior to physical synthesis, as discussed in Section IV.

## VI. CONCLUSION

New factorized carry lookahead adders viz. FRCLA and FBCLA were presented corresponding to RCLA and BCLA architectures. Based on the simulation results obtained for 32- and 64-bit additions, it is observed that the FRCLA is better optimized than the other CLAs in terms of power and energy.

## REFERENCES

- [1] B. Parhami, *Computer Arithmetic: Algorithms and Hardware Designs*, Oxford University Press, New York, 2000.
- [2] G.A. Ruiz, "New static multi-output carry lookahead CMOS adders," *IEE Proc. Circuits, Devices and Systems*, vol. 144, no. 6, pp. 350-354, 1997.
- [3] J.B. Kuo, H.J. Liao, H.P. Chen, "A BiCMOS dynamic carry lookahead adder circuit for VLSI implementation of high-speed arithmetic unit," *IEEE Jour. Solid-State Circuits*, vol. 28, no. 3, pp. 375-378, 1993.
- [4] C.-C. Wang *et al.*, "A low power high-speed 8-bit pipelining CLA design using dual-threshold voltage domino logic," *IEEE Trans. VLSI Systems*, vol. 16, no. 5, pp. 594-598, 2008.
- [5] G. Yang *et al.*, "A 32-bit carry lookahead adder using dual-path all-N logic," *IEEE Trans. VLSI Systems*, vol. 13, no. 8, pp. 992-996, 2005.
- [6] R. Zlatanovici, S. Kao, B. Nikolic, "Energy-delay optimization of 64-bit carry-lookahead adders with a 240ps 90nm CMOS design example," *IEEE Jour. Solid-State Circuits*, vol. 44, no. 2, pp. 569-583, 2009.
- [7] A. Blotti, R. Saletti, "Ultra low-power adiabatic circuit semi-custom design," *IEEE Trans. VLSI Systems*, vol. 12, no. 11, pp. 1248-1253, 2004.
- [8] J. Lim, D.-G. Kim, S.-I. Chae, "A 16-bit carry-lookahead adder using reverse-energy recovery logic for ultra-low-energy systems," *IEEE Jour. Solid-State Circuits*, vol. 34, no. 6, pp. 898-903, 1999.
- [9] A. Morgenshtein *et al.*, "Full-swing gate diffusion input logic – case-study of low power CLA adder design," *Integration*, vol. 47, pp. 62-70, 2014.
- [10] H. Cho, E.E. Swartzlander, "Adder designs and analyses for quantum-dot cellular automata," *IEEE Trans. Nanotechnology*, vol. 6, pp. 374-383, 2007.
- [11] J.F. Lope *et al.*, "Pipelined GaAs carry lookahead adder," *Electronics Letters*, vol. 34, no. 18, pp. 1732-1733, 1998.
- [12] A.H. Shaloot, A.H. Madian, "Memristor based carry lookahead adder architectures," *Proc. 55<sup>th</sup> IEEE MWSCAS*, pp. 298-301, 2012.
- [13] P. Dutta *et al.*, "Mach-Zehnder interferometer based all optical reversible carry-lookahead adder," *Proc. IEEE ISVLSI*, pp. 412-417, 2014.
- [14] Y. Sun, V. Kursun, "Low-power and compact NP dynamic CMOS adder with 16nm carbon nanotube transistors," *Proc. IEEE ISCAS*, pp. 2119-2122, 2013.
- [15] D. Sacchetto *et al.*, "Design aspects of carry lookahead adders with vertically-stacked nanowire transistors," *Proc. IEEE ISCAS*, pp. 1715-1718, 2010.
- [16] A.R. Omondi, *Computer Arithmetic Systems: Algorithms, Architecture and Implementations*, Prentice-Hall International Ltd, UK, 1994.
- [17] M.D. Ercegovac, and T. Lang, *Digital Arithmetic*, Morgan Kaufmann Publishers, California, USA, 2004.
- [18] Synopsys SAED\_EDK32/28\_CORE Databook, Revision 1.0.0, 2012.
- [19] P. Balasubramanian, N. Mastorakis, "Performance comparison of carry-lookahead and carry-select adders based on accurate and approximate additions," *Electronics*, vol. 7, no. 12, Article #369, pages 12, 2018.