Design and Implementation of 64-bit Carry Lookahead Adders Using Fixed and Variable Stage Structure

Rasmita Barik
Dept. of Instrumentation and Electronics Engineering
College of Engineering& Technology
Bhubaneswar, Odisha
rasmitabarik123@gmail.com

Soumyashree Mangaraj
Dept. of Instrumentation and Electronics Engineering
College of Engineering& Technology
Bhubaneswar, Odisha
Soumya.mangaraj9@gmail.com

Abstract— **Adders are basic integral part of arithmetic circuits. The adders have been realized with two styles: fixed stage and variable stage size. This paper presents the correlation investigation of execution examination of 64-bit Carry Lookahead Adders utilizing conventional and hierarchical structure styles with fixed stages and variable stages. We utilize different diverse parameter to evaluate conventional carry lookahead adder (CLA) and hierarchical carry lookahead adder (HCLA) and variable stage carry lookahead adder. Our outline is actualized into Zedboard Xilinx Zynq XC7Z020-1CLG484. Our intrigued of investigation are delay, area, and power. In this paper we show conventional CLA required small area using radix-2, while in hierarchical CLA delay is diminished to a great extent. Furthermore, we demonstrated variable stages CLA would be able to tradeoff between the area, delay and power.**

Keywords—*Conventional Carry Lookahead Adder (CLA) Hierarchical Carry lookahead Adder (HCLA), Varible stage CLA(CLA-V). Varible stage HCLA (HCLA-V), VLSI design and FPGA implementation*

I. Introduction

Digital adder circuits are considered as a most commonly used hardware in any digital equipment for various applications. Binary adder most abundantly used in DSP processors, embedded processors, and high-performance processor rely on the adder circuit in order to perform algorithm operations in their ALU unit. Major constraints these processor are consumption of power and speed of operation. However, optimization of an ideal adder can be done in three areas are power consumption, delay and area. In the past, certain parameters involving power reduction, time delay and die area have to be taken into consideration in designing of these processors [1], [2], [3]. In addition, today's processors execute millions of instructions in which speed of operation becomes one of the major constraints of processor design. Here we formulate the tradeoff between delay, area and power to maintain the requirement of our high efficient digital device.

II. Motivation

There are number of researchers purposed the theory for Carry Lookahead Adder (CLA) using both conventional structure and hierarchical structure [4], [5], [6], [7], [8], [9]. Most of the previous suggestions of enhancing CLA performance depends on the implementing CLA adder using different logic gate styles [10], [11], [12], [13], [14], various tree structures [15], [16], [17], or enhancing performance using optimizing techniques [18], [19]. In this work, a general methodology is developed for constructing 64-bit Variable stage CLA and Variable stage HCLA using number of modules (m) where there are no restrictions for radix-n in each module. Unlike the traditional approach that the module of CLA be 4 we provide the variable stages input.

- We introduced and compared 64-bit CLA, HCLA, CLA-V and HCLA-V using different radix to reach the best value of n in order to reduce delay, area and power.
- We showed that using radix-2 CLA has better performance than commonly used radix-4, as well as better performance than HCLA.
- We showed that conventional CLA required small area using radix-2, while in hierarchical CLA delay is reduced largely.
- We introduced our proposed variable stages CLA adder which is better tradeoff between area, delay and power.

The structure of the paper is organized as follows: In section III, we discuss 64-bit CLA and shows the different implementation using identical radix-n. Section IV, shows 64-bit HCLA basic modules design and implementation using radix-*n*. Section V, show the 64-bit variable stages CLA & HCLA using non-identical radix-n. Results and discussion are reported in Section VI. Finally, we offer the conclusion in section VII.

III. Design And Implementation Of 64 Bit CLA Adder

For n-bit CLA, the two *n*-bit inputs $a[n-1:0]$ and $b[n-1:0]$ to be added are used to generate the carry propagate $p[n-1:0]$ and carry generate $g[n-1:0]$ signals according to the equation:

$$p_i = a_i \oplus b_i \quad (1)$$

$$g_i = a_i . b_i \quad (2)$$

ISBN: 978-1-5386-3476-9/18/$31.00 ©2018 IEEE
16th & 17th Mar 2018

Structure of 64-bit carry lookahead adder in the Figure.1 contains Partial Full Adder (PFA) which is the responsible for generate and propagate the carry to the CLA module. Input carry for the Partial full adder (PFA) is generated by carry generator logic. Figure1 shows radix-2 conventional CLA structure style.

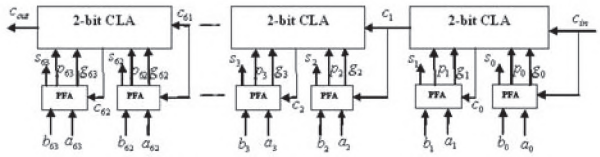

Figure 1. 64 bit CLA using radix 2

Figure 2 shows the *n*-bit CLA basic module which accepts two *n*-bit input signals *p* and *g*, and the carry-in signal C_{in} and produces $n + 1$ bits carry signal.

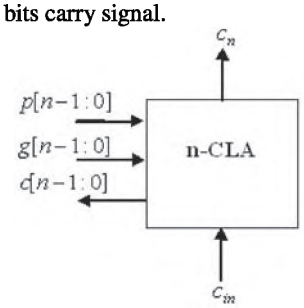

Figure 2. *n*-bit CLA basic module

Boolean expression for carry signals C_0 to C_3 are:

$$c_0 = c_{in}$$
$$c_1 = g_0 + p_0.c_0$$
$$c_2 = g_1 + p_1.g_0 + p_1.p_2.c_0$$
$$c_3 = g_2 + p_2.g_1 + p_2.p_1.g_0 + p_2.p_1.p_0.c_0$$

In general, we can write the carry at bit *i* as:

$$c_i = c_{in} \prod_{j=0}^{i-1} p_j + \sum_{j=0}^{i-1} g_j \prod_{k=j+1}^{i-1} p_k; 0 \le i \le n \quad (3)$$

The sum signals are obtained as:

$$s = p_i \oplus c_i ; \ 0 \le i < N \quad (4)$$

TABLE I
64 BIT CLA USING RADIX -N

No. of modules (m)	*Radix-n in Each module*
2	32
4	16
8	8
16	4

In the implementation of *N*-bit CLA using conventional structure can be obtained according to the equation N = *m* x *n*, where *m* is the number of modules and *n* is the radix in each module. The lookahead modules compute the carry bits based on generate and propagate values they receive as discuss in the Boolean equation. Our studies based on the different combination of m and n for 64-bit CLA.

IV. DESIGN AND IMPLEMENTATION OF 64-BIT HCLA

In the design of n-bit HCLA module that accepts *n*-bit $p[n-1:0]$ and $g[n-1:0]$ signals and produces two signals: group carry propagate p_{out} and group carry generate g_{out} as discussed in [20]. The propagation carry and carry generate signals are given by:

$$P_{out} = \prod_{j=0}^{n-1} p_j \quad (5)$$

$$g_{out} = \sum_{i=0}^{n-1} g_i \ \prod_{j=i+1}^{n-1} P_j \quad (6)$$

The *Cn* of *n*-HCLA can be obtained through a small circuit which is given by:

$$c_n = c_{in} . p_{out} + g_{out} \quad (7)$$

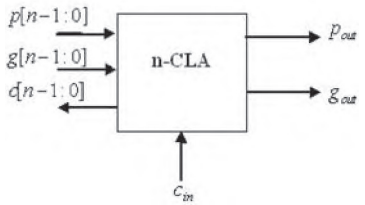

Figure 3. n-bit HCLA basic module

For the N-bit HCLA can be obtained by multiplying the radix-n in the first level (h_0) by the radix (n) in the second levels (h_1) according to the equation: N = n(h_0) x n(h_1) Figure 4 shows two levels implementation of 64-bit HCLA using identical radix-8 in both the level. The two signals *g* and *p* indicate to carry generate and carry propagate respectively. These signals are still a single bit each they are not a vector. The carry out signal C_{64} is obtained at the top level of the hierarchy based on equation 7.

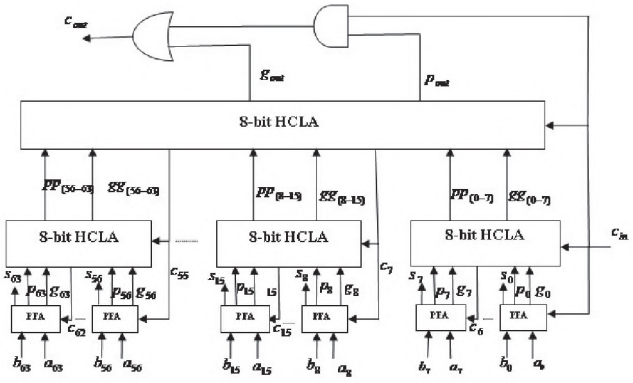

Figure 4. 64-bit HCLA using identical radix-8

We construct 64-bit non-identical radix-n deals with the case when the radix-n in level '0' and level '1' are not the same. The first level has four modules based on the radix-16 each. The second level has one module based on the radix-4. As shown in figure 5, 64-bit HCLA using non identical radix-n in their various hierarchies.

ISBN: 978-1-5386-3476-9/18/$31.00 ©2018 IEEE
16th & 17th Mar 2018

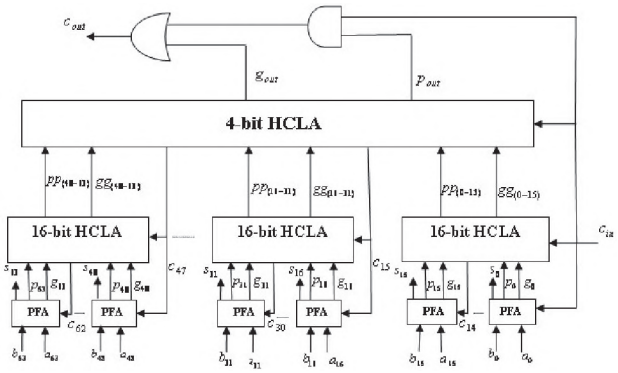

Figure 5. 64-bit HCLA using non identical radix-n

Tables II illustrate two HCLAs implementation using non identical radix-n. The first row indicates the number of radix in first levels while the second row shows the number of radix in second level. It can be seen that we need three modules to design 64- bit HCLA using radix-32 in the first level and radix-2 in the second level. In contrast, we need five modules to design 64- bit HCLA using radix-16 in the first level and radix-4 in the second level as shown in Figure 5.

TABLE II
2 LEVEL 64-BIT HCLA USING NON IDENTICAL RADIX-N

Level (h)	*Radix-n in each level*	
h_0	32	16
h_1	2	4
No. of module	3	5

V. DESIGN OF VARIABLE CLA AND HCLA

The conventional variable stage CLA is shows in figure 6 and the hierarchical variable stage CLA is shown in figure 7.
The proposed adder has fifteen stages where first and last stage are of 1 bit each, it keeps increasing steadily till the middle stage which is the bulkiest and hence this is the nucleus stage of circuit.

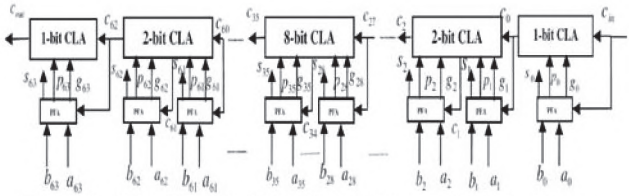

Figure 6. 64-bit Variable Stage CLA using non identical radix-n

The CLA-V is designed to have fifteen stages, starting from 1-bit, consecutive stages are incremented by 1-bit till the middle stage that is of 8 bits. This is the nucleus stage. After the nucleus stage the consecutive stages are decreased by 1-bit till the last stage. In this configuration the middle stage is the bulkiest whereas the first and last stages are the smallest.

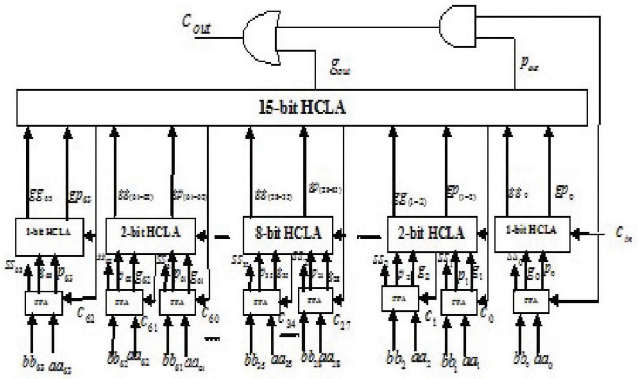

Figure 7. 64-bit Variable Stage HCLA using non identical radix-n

VI. IMPLEMENTATION RESULTS AND DISCUSSION

In this paper, the hardware execution of 64-bit variable stage and fixed stage CLA and HCLA using various radix-*n* has been carried out. We utilized Verilog code for composing RTL schematic and Xilinx Vivado tool to simulate and synthesize the design. The simulation assists to verify the design and the synthesis report shows the results of delay, area and power. The outcomes demonstrate an examination between conventional, hierarchical structures of CLA, variable stages CLA and variable stage HCLA. Tables III illustrate 64-bit CLA implementation results using different radix-n. The design implemented into Zedboard Xilinx Zynq XC7Z020-1CLG484 to estimate the worst-case delay (*T*), area (*A*), power. Area estimation is based on the number of slices utilized in each design.

TABLE III
64 BIT CONVENTIONAL FIXED STAGE CLA IMPLEMENTATION RSULTS OF DELAY, AREA AND POWER

M	Radix-n	T(ns)	A	Power
32	Radix-2	19.680	109	46.981
16	Radix-4	16.739	120	47.219
8	Radix-8	13.444	137	47.519
4	Radix-16	11.332	201	48.316
2	Radix-32	9.720	184	49.812

It is noticed in Table III, that increasing the value of *n* leads to the worst-cases in terms of area and power. Radix-2 produces smaller area and power as shown in Table III. Increasing the value of n in CLA design produces large area and power due to utilized resources and routing complexity. Exceptionally, Radix-32 produces small area and power as compared to radix-16.
Table IV shows the simulation results of 64-bit HCLA using two levels implementation. It can be shown that 64-bit HCLA designed with identical radix-8 produces the least delay. The HCLA using non-identical radix such as radix-32 in level '0' and radix-2 in level '1' has the more delay.

ISBN: 978-1-5386-3476-9/18/$31.00 ©2018 IEEE
16th & 17th Mar 2018

TABLE IV
64 BIT FIXED STAGE HCLA IMPLEMENTATION RESULT OF DELAY, AREA AND POWER

Radix-n	T(ns)	A	Power
Radix-2	11.688	151	48.020
Radix-4	10.552	178	48.140
Radix-8	10.330	193	47.865
Radix-16	10.374	203	48.154
Radix-32	13.589	181	47.996

TABLE V
VARIABLE STAGES CLA IMLEMENTATION RESULT

N-Bit CLA	Module (M)	Area	T(ns)	Power
64	15	136	14.520	47.025

In Table V, it is shows the result of delay, area and power of Variable stage CLA where we get better tradeoff between area, time and power as compare to conventional.

TABLE VI
VARIABLE STAGES HCLA IMLEMENTATION RESULT

N-Bit CLA	Module (M)	Area	T(ns)	Power
64	16	191	10.233	47.468

In Table VI, it is shows the result of area and delay of Variable stage HCLA. It also gives lower delay time as compare to fixed stage CLA and HCLA.

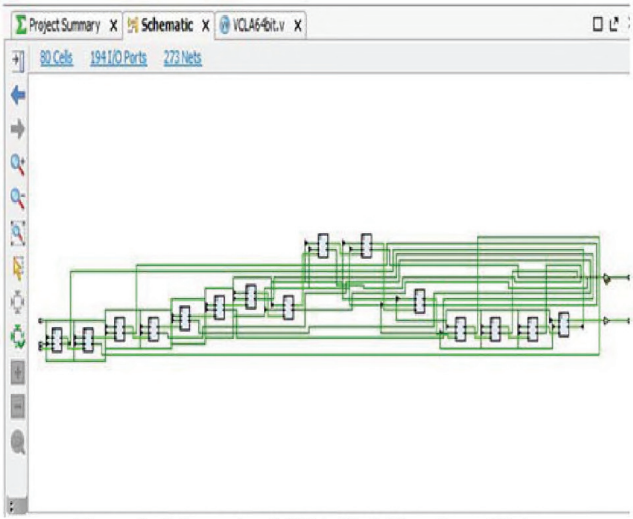

Figure 8. Schematic of 64-bit Variable s Stage CLA

Figure 8, shows the RTL schematic of the 64bit variable stage CLA, where first and last stages are of 1 bit each stage. Number Bit will be increase to the middle Stage. It is symmetrical from the middle stage. Hence the middle level is nucleus stage and it is bulkiest part of the device.

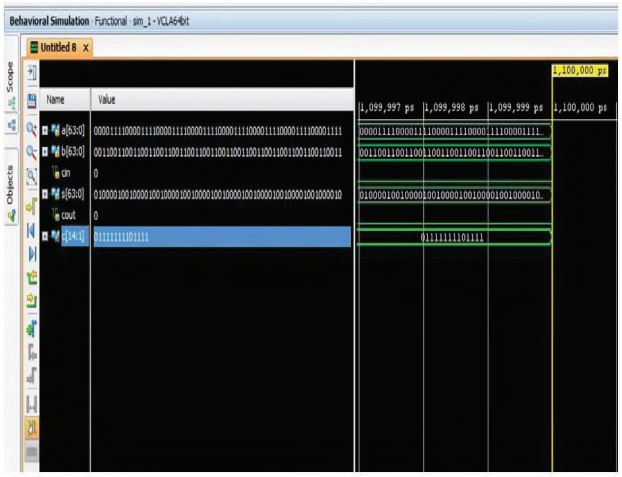

Figure 9. Simulation result of 64-bit Variable stage CLA

Figure.9, shows the behavioral simulation output of the 64bit Variable stage CLA. Where a, b are the 64 bit input and C is the carry-in input. S is the 64-bit sum of the CLA and c_{out} is the carry-out output.

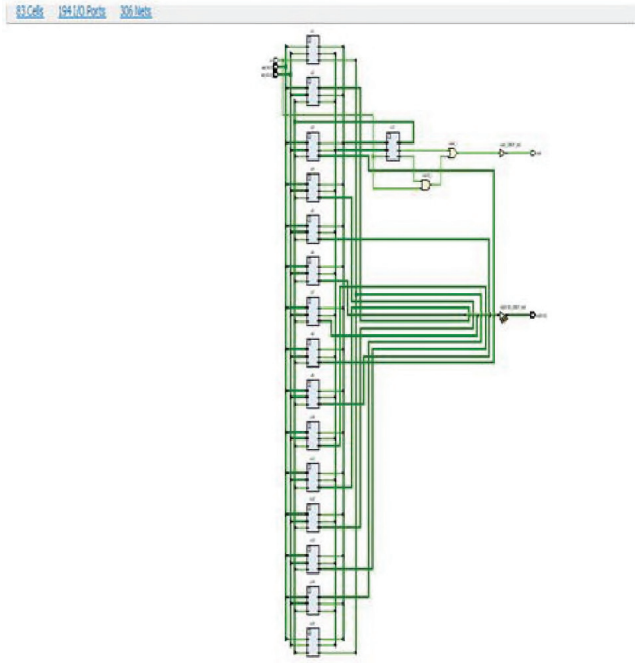

Figure 10. RTL Schematic of 64-bit Variable Stage HCLA

Figure 10, shows the RTL schematic of the 64bit variable stage HCLA. Where first and last stages are of 1 bit each, it keeps increasing steadily till the middle stage which is the bulkiest and hence, it is the nucleus stage. After the nucleus stage the consecutive stages are decrease by one bit till the last stage.

ISBN: 978-1-5386-3476-9/18/$31.00 ©2018 IEEE
16th & 17th Mar 2018

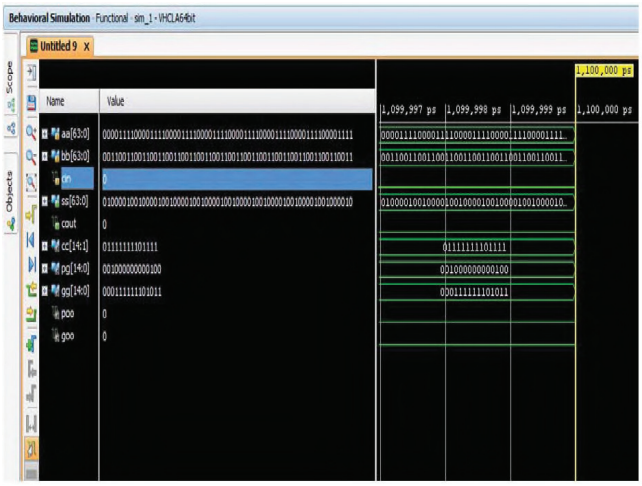

Figure 10. Simulation result of 64-bit Variable stage HCLA

Figure 11, shows the behavioral simulation output of the 64-bit Variable stage HCLA. Where aa, bb are the 64 bit input and C_{in} is the carry-in input, SS is the 64-bit sum of the CLA and C_{out} is the carry-out output.

The implementation results in Table III, Table IV, Table V and Table VI can be summarized as following:

- Increasing the value of n doesn't contribute to improve the performance of CLA design
- CLA using radix-2 has lesser area as compared to other style structure.
- HCLA using radix-8 has always less delay as compared to any other radix.
- Increasing the value of n leads to a huge area for both CLA and HCLA design.
- Variable stages CLA are tradeoff between area and power.

VII. CONCLUSION

In this work, we designs and execute the CLA and HCLA for fixed and variable stages. Our investigation focus on 64 bit CLA and HCLA execution for various cases. Our design targeted Zedboard Xilinx Zynq XC7Z020-1CLG484 to estimate the delay, area and power. The outcomes demonstrated that the variable stages CLA better execution regarding area, delay and power. By utilizing variable radix stages CLA, we enhance the area and considerable level of delay. It provides the solution for both faster operation and lesser area requirement. It is also minimize the power requirement for variable stages carry look adder. Results of the paper conclude that the variable stages CLA has better performance than fixed-stage radix-n CLA and HCLA. A little tradeoff between area, delay and power can improve the overall performance.

REFERENCES

[1] Abdulmajeed Alghamdi and Fayez Gebali, "Performance analysis of 64 bit Carry Lookahead Adders usingconventional and hierarchical structur e style" Communications, Computers and Signal Processing (PACRIM), 2015, pp. 80-83.

[2] H. Elmiligi, M. W. El-Kharashi, and F. Gebali, "Power consumption of 3d networks-on-chips: Modeling and optimization," Microprocessors and Microsystems, vol. 37, no. 6, pp. 530–543, 2013.

[3] B. H. Meyer, J. J. Pieper, J. M. Paul, J. E. Nelson, S. M. Pieper, and A. G. Rowe, "Power-performance simulation and design strategies for single-chip heterogeneous multiprocessors," Computers, IEEE Transactions on, vol. 54, no. 6, pp. 684–697, 2005.

[4] R. Uma, V. Vijayan, M. Mohanapriya, and S. Paul, "Area, delay and power comparison of adder topologies," International Journal of VLSI design & Communication Systems (VLSICSj Vo1. 3, No. 1, 2012.

[5] M. H. Hjakazemi and A. Baniasadi, "An alternative hybrid poweraware adder for high-performance processors," Journal of Low Power Electronics, vol. 10, no. 1, pp. 38–44, 2014.

[6] R. Singh, P. Kumar, and B. Singh, "Performance analysis of fast adders using vhdl," in Advances in Recent Technologies in Communication and Computing, 2009. ARTCom'09. International Conference on. IEEE, 2009, pp. 189–193.

[7] M. Hajkazemi, A. Haghdoost, and A. Baniasdi, "Reconfiguring the carry look-ahead adder using application behavior in embedded processors," in Electrical Engineering/Electronics Computer Telecommunications and Information Technology (ECTI-CON), 2010 International Conference on. IEEE, 2010, pp. 183–187.

[8] P. Gurjar, R. Solanki, P. Kansliwal, and M. Vucha, "Vlsi implementation of adders for high speed alu," in India Conference (INDICON), 2011 Annual IEEE. IEEE, 2011, pp. 1–6.

[9] L. M. Kalyani Garimella, S. R. Sudha Garimella, K. Duda, and E. Fetzer, "New generation carry look twice-ahead adder cl2a and carry look thrice-ahead adder cl3a," in Circuits and Systems (MWSCAS), 2013 IEEE 56th International Midwest Symposium on. IEEE, 2013, pp. 1387–1390.

[10] R. Kumar and S. Dahiya, "Performance analysis of different bit carry look ahead adder using vhdl environment," vol. 2, 2013.

[11] J. Samanta, M. Halder, and B. P. De, "Performance analysis of high speed low power carry look-ahead adder using different logic styles," International Journal of Soft Computing and Engineering (IJSCE) ISSN, pp. 2231–2307, 2013.

[12] P. Celinski, J. F. L´opez, S. Al-Sarawi, and D. Abbott, "Low depth, low power carry lookahead adders using threshold logic," Microelectronics journal, vol. 33, no. 12, pp. 1071–1077, 2002.

[13] K. Ueda, H. Suzuki, K. Suda, H. Shinohara, and K. Mashiko, "A 64-bit carry look ahead adder using pass transistor bicmos gates," Solid-State Circuits, IEEE Journal of, vol. 31, no. 6, pp. 810–818, 1996.

[14] K.-H. Cheng, S.-W. Cheng, and W.-S. Lee, "64-bit pipeline carry lookahead adder using all-n-transistor tspc logics," Journal of Circuits, Systems, and Computers, vol. 15, no. 01, pp. 13–27, 2006.

[15] M. C. Osorio, C. Sampaio, A. Reis, R. P. Ribas et al., "Enhanced 32- bit carry look-ahead adder using multiple output enable-disable cmos differential logic," pp. 181–185, 2004.

[16] C. Dacheng, "Vhdl implementation of a fast adder tree," 2005.

[17] C.-C. Wang, P.-M. Lee, R.-C. Lee, and C.-J. Huang, "A 1.25 ghz 32-bit tree-structured carry lookahead adder," vol. 4, pp. 80–83, 2001.

[18] S. H. Kim and S.-K. Chin, "Formal verification of tree-structured carrylookahead adders," pp. 232–232, 1999.

[19] R. Zlatanovici, S. Kao, and B. Nikolic, "Energy–delay optimization of 64-bit carry-lookahead adders with a 240 ps 90 nm cmos design example," Solid-State Circuits, IEEE Journal of, vol. 44, no. 2, pp. 569–583, 2009.

[20] F. Gebali and A. Ibrahim, "Optimized structures of hybrid ripple carry and hierarchical carry lookahead adders," Microelectronics, 2015, in

ISBN: 978-1-5386-3476-9/18/$31.00 ©2018 IEEE
16th & 17th Mar 2018