

HIGH SPEED AND INDEPENDENT CARRY CHAIN CARRY LOOK AHEAD ADDER (CLA) IMPLEMENTATION USING CADENCE-EDA

K.Krishna Kumar¹, A.Nandha Kumar²

¹*Department of Electrical and Electronics Engineering,*

Dr.Mahalingam College of Engineering and Technology, Pollachi (India)

²*Assistant Professor, Department of Electrical And Electronics Engineering ,*

Dr.Mahalingam College of Engineering and Technology, Pollachi (India)

ABSTRACT

In this paper focuses on carry -look ahead adders have done research on the design of high-speed, low-area, or low-power adders. Addition is the fundamental operation for any VLSI processors or digital signal processing. The main objective of this paper is to reduce the propagation delay and gate count of the Carry look-Ahead Adder (CLA). Which will also reflect in the reduction of area and power of the adder module. Experimental results reveal that the proposed adders achieve delay, power and area reductions for Multi bit addition. We know that in adder circuits propagation delay is the main drawback. To overcome this drawback the independent carry technique is introduced. Here in this paper 4, 8, 16-bit adders are been designed and the gate count, power and delay are measured using CADENCE EDA, and then compared with the conventional adder.

Index Terms: Carry look-ahead (CLA) adder, VLSI, CADENCE EDA.

I. INTRODUCTION

Addition is the most commonly used arithmetic operation and also the speed-limiting element to make faster VLSI processors. As the demand for higher performance processors grows, there is a continuing need to improve the performance of arithmetic units and to increase their functionality. High-speed adder architectures include the carry look-ahead (CLA) adders, carry-skip adders, carry-select adders, Conditional sum adders and combinations of these High-speed adders based on the CLA principle remain dominant, since the carry delay can be improved by calculating each stage in parallel.

The CLA algorithm was first introduced in 1958, and several variants have been developed. The Manchester carry chain (MCC) is the most common dynamic (domino) CLA adder architecture with a regular, fast, and simple structure adequate for implementation in VLSI. The recursive properties of the carries in MCC have enabled the development of multioutput domino gates, which have shown area-speed improvements with respect to single-output gates.

In this brief, the carry will be generated independently without the use of previous carries. In previous work of this paper the carry chain has been split into even and odd carry chains. The even and odd carries of this adder are computed in parallel by two independent 4-bit carry chains. And in this paper the delay for 4 and 8-bit will

be higher when compared to the conventional adder but for 16-bit and more than that delay will be reduced drastically.

This paper is organized as follows. In Section II, preliminary concepts of the double carry chain MCC adders are given. In Section III, the proposed independent carry adder is presented. In Section IV, comparisons among the proposed design and conventional MCC topologies in the open literature are given. Finally, in Section V, the conclusions are drawn.

II. PRELIMINARY CONCEPTS OF DOUBLE CARRY CHAIN MCC ADDERS

MCC adders can efficiently be designed in CMOS logic. As mentioned previously, due to technological constraints, the length of their carry chains is limited to 4 bits. However, these 4-bit adder blocks are used extensively in the literature [2] and [12] in the design of wider adders.

In the following, we propose the design of an 8-bit adder module which is composed of two independent carry chains. These chains have the same length (measured as the maximum number of series-connected transistors) as the 4-bit MCC adders. According to our simulation results, the use of the proposed 8-bit adder as the basic block, instead of the 4-bit MCC adder, can lead to high-speed adder implementations.

The derived here carry equations are similar to those for the Ling carries proposed in [4]. The derived carry equations allow the even carries to be computed separately of the odd ones. This separation allows the implementation of the carries by two independent 4-bit carry chains; one chain computes the even carries, while the other chain computes the odd carries. In the following, the design of the proposed 8-bit MCC adder is analytically presented.

2.1. Even Carry Computation

For $i = 0$ and $z_0 = t_0$, from relation (1), we get that $c_0 = g_0 + t_0 \cdot c_{-1}$. Since the relation $g_i = g_i \cdot t_i$ holds, we get that $c_0 = t_0 \cdot (g_0 + c_{-1}) = t_0 \cdot h_0$, where $h_0 = g_0 + c_{-1}$ is the new carry. From relation (2), for $i = 2$ and $z_i = p_i$, we get that

$$c_2 = g_2 + p_2 g_1 + p_2 p_1 g_0 + p_2 p_1 p_0 c_{-1}.$$

Since $g_i + p_i \cdot g_{i-1} = g_i + t_i \cdot g_{i-1}$ and $p_i = p_i \cdot t_i$, we have

$$c_2 = t_2 (g_2 + g_1 + p_2 p_1 g_0 + p_2 p_1 p_0 c_{-1}) = t_2 (g_2 + g_1 + p_2 p_1 t_0 (g_0 + c_{-1})) = t_2 \cdot h_2$$

where

$$h_2 = g_2 + g_1 + p_2 p_1 t_0 (g_0 + c_{-1}) \text{ is the new carry.}$$

In the same way, the new carries for $i = 4, 6$ are computed as

$$h_4 = g_4 + g_3 + p_4 p_3 t_2 (g_2 + g_1 + p_2 p_1 t_0 (g_0 + c_{-1}))$$

$$h_6 = g_6 + g_5 + p_6 p_5 t_4 \times (g_4 + g_3 + p_4 p_3 t_2 (g_2 + g_1 + p_2 p_1 t_0 (g_0 + c_{-1}))).$$

Then, the following equations are derived for the new

carries for even values of i :

$$h_2 = G_2 + P_2 G_0$$

$$h_4 = G_4 + P_4 G_2 + P_4 P_2 G_0$$

$$h_6 = G_6 + P_6G_4 + P_6P_4G_2 + P_6P_4P_2G_0$$

2.2. Odd Carry Computation

The new carries for the odd values of i are computed according to the aforementioned methodology proposed for the even carries as follows:

$$h_1 = g_1 + g_0 + p_1p_0c^{-1}$$

$$h_3 = g_3 + g_2 + p_3p_2t_1 (g_1 + g_0 + p_1p_0c^{-1})$$

$$h_5 = g_5 + g_4 + p_5p_4t_3 (g_3 + g_2 + p_3p_2t_1 (g_1 + g_0 + p_1p_0c^{-1}))$$

$$h_7 = g_7 + g_6 + p_7p_6t_4$$

$$\times (g_5 + g_4 + p_5p_4t_3$$

$$\times (g_3 + g_2 + p_3p_2t_1 (g_1 + g_0 + p_1p_0c^{-1}))) .$$

While for odd values of i , the equations for the new carries are rewritten as follows:

$$h_1 = G_1 + P_1c^{-1}$$

$$h_3 = G_3 + P_3G_1 + P_3P_1c^{-1}$$

$$h_5 = G_5 + P_5G_3 + P_5P_3G_1 + P_5P_3P_1c^{-1}$$

$$h_7 = G_7 + P_7G_5 + P_7P_5G_3 + P_7P_5P_3G_1 + P_7P_5P_3P_1c^{-1}.$$

From the aforementioned equations, it is evident that the groups of even and odd new carries can be computed in parallel by different carry chains in multioutput domino CMOS logic. The new generate and propagate signals G_i and P_i can be easily proven to be mutually exclusive, avoiding false node discharges. Between the new and the conventional carries, $c_i^{-1} = t_i^{-1} \cdot h_i^{-1}$ holds; therefore, the sum bits are computed as $s_i = p_i \oplus (t_i^{-1} \cdot h_i^{-1})$. According to [4], the computation of the sum bits can be performed as follows:

$$s_i = h_i^{-1} \cdot p_i + h_i^{-1} \cdot (p_i \oplus t_i^{-1})$$

$$\text{for } i > 0, \text{ while } s_0 = p_0 \oplus c^{-1}.$$

The above Relation can be implemented using a $2 \rightarrow 1$ multiplexer that selects either p_i or $p_i \oplus t_i^{-1}$ according to the value of h_i^{-1} . Taking into account that an XOR gate introduces equal delay with a $2 \rightarrow 1$ multiplexer and both terms p_i and $p_i \oplus t_i^{-1}$ are computed faster than h_i , then no extra delay is introduced by the use of the proposed carries for the computation of the sum bits according to above equation.

For the implementation of the sum signals, the domino chain is terminated, and static CMOS technology is used for the $p_i \oplus t_i^{-1}$ gate and the final $2 \rightarrow 1$ multiplexer. An efficient static CMOS implementation of the $2 \rightarrow 1$ multiplexer is used for Sum bit implementation.

2.3. MCC Design Issues

To evaluate the speed performance of the proposed (PROP) design over the conventional (CONV) one, 8-, 16-, 32-, and 64-bit adders have been designed according to the even and odd carry chain principle respectively, and simulated using SPECTRE in a standard 90-nm CMOS technology ($V_{DD} = 1$ V). The conventional 8-, 16-, 32-, and 64-bit MCC adders are designed by cascading two, four, eight, and sixteen 4-bit MCC adder modules,

respectively. The proposed 16-, 32-, and 64-bit MCC adders are designed by cascading two, four, and eight of the proposed 8-bit MCC adder modules, respectively.

III. PROPOSED WORK

In this work i have proposed the new independent carry technique which will not require the previous carry bit to generate the current carry. While it will generate all the carry bits parallel. The proposed Carry Equations for 4-bit are

- $C(0) = e(0) \text{ AND } p(0);$
- $C(1) = (e(1) \text{ AND } p(1) \text{ OR } p(1) \text{ AND } g(0) \text{ or } k(1);$
- $C(2) = (e(2) \text{ AND } p(2) \text{ OR } p(2) \text{ AND } g(1) \text{ or } k(2);$
- $C(3) = (e(3) \text{ AND } p(3) \text{ OR } p(3) \text{ AND } g(2) \text{ or } k(3);$

INTERMEDIATE TERMS

- $e(i) = a(i) \text{ XNOR } b(i);$
- $k(i) = p(i) \text{ and } \dots\dots\dots p(0);$

PROPAGATE AND GENERATE TERMS

- $p(i) = a(i) \text{ OR } b(i);$
- $g(i) = a(i) \text{ AND } b(i);$

SUM BIT IMPLEMENTATION

- $S(i) = a(i) \text{ xor } b(i) \text{ xor } c(i-1);$

IV. RESULTS AND COMPARISION.

TABLE I

CLA PARAMETRIC COMPARISION

n-bit	Proposed			Conventional		
	Delay (ps)	Power (nW)	Area (μm^2)	Delay (ps)	Power (nW)	Area (μm^2)
4-bit	128.25	540.47	32	97.3	663.41	38
8-bit	128.25	1125.4	64	120.21	1533.2	77
16-bit	128.25	2635.1	144	168.57	3765.5	175

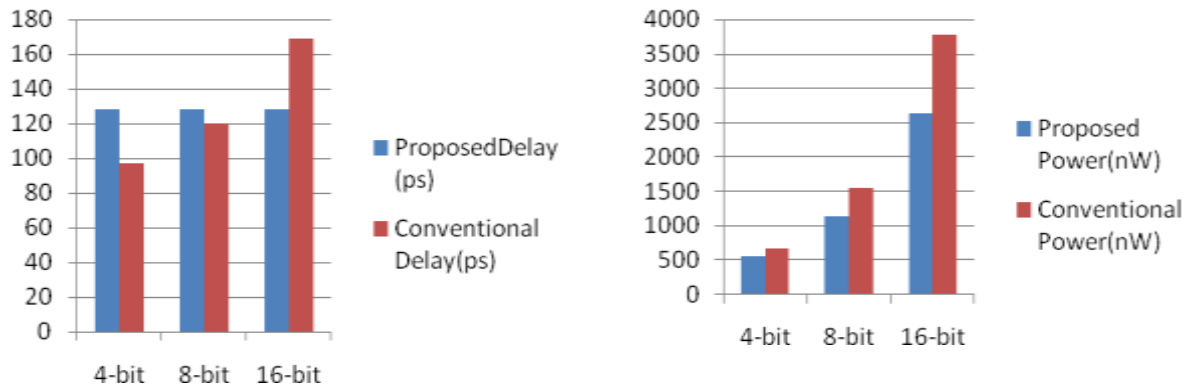


FIGURE 1(a). Delay Comparison Chart FIGURE 1(b). Power Comparison Chart

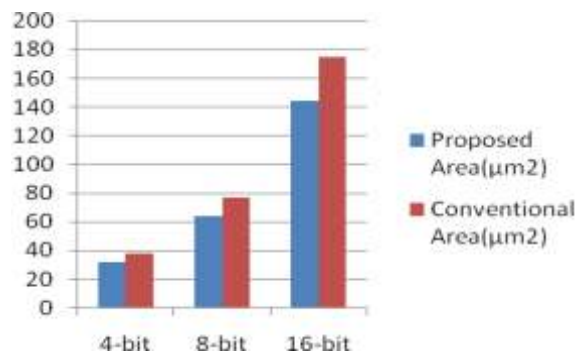


FIGURE 1(c). Area Comparison Chart

TABLE II

Percentage Comparison of Proposed and Conventional Adder Parameters

n-bit	Delay(%)	Power(%)	Area(%)
4-bit	-32.8	18.5	15.78
8-bit	-6.68	26.59	16.88
16-bit	23.91	30.01	17.71

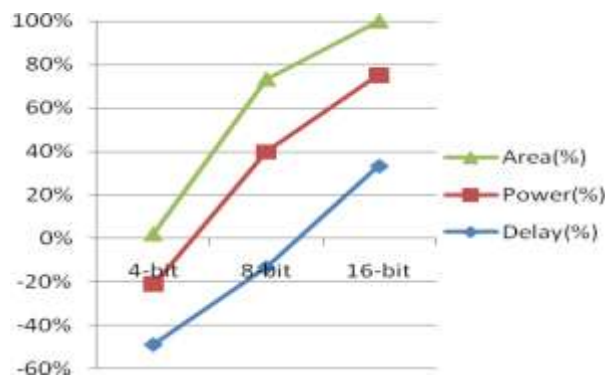


FIGURE 2. Percentage Improvement of Proposed Technique Over Conventional

V. CONCLUSION

The Independent Carry technique is an efficient approach to construct Fast CLA adders. In this work, I have presented a new independent carry technique, which does not need to wait for the previous carry to be calculated. In this way the delay has been reduced for the higher bit addition. Here the small drawback is the delay reduction will be efficient for 16-bit and larger than 16-bit addition and for some combinations of input carry will not propagate properly. And the proposed work will also reduce area and power of the adder.

REFERENCES

- [1] Scott Hauck, *Member, IEEE*, Matthew M. Hosler, and Thomas W. Fry "High-Performance Carry Chains for FPGA's" IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS, VOL. 8, NO. 2, APRIL 2000.
- [2] Yukinori Ono, Member et al IEEE, "Binary Adders of Multigate Single-electrontransistors: Specific Design usingpass-Transistor Logic",IEEE TRANSACTIONSON NANOTECHNOLOGY, VOL. 1, NO. 2, JUNE 2002.
- [3] M. Alioto and G. Palumbo "A Simple Strategy for Optimized Design of One-Level Carry-Skip Adders" IEEE transactions on circuits and systems i: fundamental theory and applications, vol. 50, no. 1, january 2003.
- [4] Giorgosdimitrakopoulos and dimitrisnikolos, Member, IEEE "High- Speed Parallel-Prefix VLSI Ling Adders" IEEE transactions on computers, vol. 54, no. 2, february 2005.
- [5] Jung-Yup Kang, Member, IEEE, and Jean-Luc Gaudiot, Fellow, IEEE" A Simple High-Speed Multiplier Design" IEEE TRANSACTIONS ON COMPUTERS, VOL. 55, NO. 10, OCTOBER 2006.
- [6] Fatemehkashfi, and Nasser Masoumi "Optimization of Speed and Power in a 16-Bit Carry Skip Adder in 70nm Technology ©2006 IEEE.
- [7] Daniela Elena Popescu and corneliupopescu "Efficient Algorithms For Testing The Two-Level Carry Skip Adder" ©2006 IEEE.
- [8] A.Bharathi , K.Manikandan , K.Rajasri , P.Santhini"High Speed Multioutput 128bit carrylookahead Adders Using Domino Logic"Vol. 3, Issue 10, October2014.
- [9] Costas et al, Member IEEE,"new high speed multi output carry look –ahead adders"IEEE transactions on circuits and systems—II: express briefs, vol. 60, no. 10, october 2013
- [10] Itamar Levi, Ori Bass, asafaizerman, Alexander Belenky and Alexander Fish, memberieee "High Speed Dual Mode Logic Carry Look Ahead Adders" Low Power Circuits & Systems Lab, the VLSI Systems Center, Ben-Gurion University of the Negev, Israel 2012 IEEE.
- [11] Pierce Chuang et alFellow, IEEE "A Low-Power High-Performance Single-Cycle Tree-Based 64-Bit Binary Comparator"IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—II: EXPRESS BRIEFS, VOL. 59, NO. 2, FEBRUARY 2012.
- [12]B. Parhami, Computer Arithmetic, Algorithms and Hardware.New York, USA: Oxford Univ. Press, 2000.

BIOGRAPHICAL NOTES

Mr K. Krishna kumaris presently pursuing M.E. Final year in Electrical and Electronics Engineering Department (Specialization in Applied Electronics) from Dr. Mahalingam College of Engineering and Technology, Pollachi (India)

Mr A. Nandha kumar is working as an Assistant Professor in Electrical and Electronics Engineering Department, Dr. Mahalingam College of Engineering and Technology, Pollachi (India) and presently pursuing Ph. D. from Anna University Chennai (India).