New High-Speed Multioutput Carry Look-Ahead Adders

Costas Efstathiou, Zaher Owda, and Yiorgos Tsiatouhas, *Member, IEEE*

***Abstract*—In this brief, an efficient implementation of an 8-bit Manchester carry chain (MCC) adder in multioutput domino CMOS logic is proposed. The carries of this adder are computed in parallel by two independent 4-bit carry chains. Due to its limited carry chain length, the use of the proposed 8-bit adder module for the implementation of wider adders leads to significant operating speed improvement compared to the corresponding adders based on the standard 4-bit MCC adder module.**

***Index Terms*—Carry look-ahead (CLA) adders, Manchester carry chain, multioutput domino logic.**

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 structures [1]–[4]. 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 [5], 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 [6], [7]. 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, a new 8-bit carry chain adder block in multioutput domino CMOS logic is proposed. The even and odd carries of this adder are computed in parallel by two independent 4-bit carry chains. Implementation of wider adders based on the use of the proposed 8-bit adder module shows significant operating speed improvement compared to their corresponding adders based on the standard 4-bit MCC adder module.

This brief is organized as follows. In Section II, preliminary concepts on the domino design of MCC adders are given. In Section III, the architecture of the proposed double carry chain 8-bit MCC adder is presented. In Section IV, comparisons among the proposed MCC design and two conventional MCC topologies in the open literature are given. Finally, in Section V, the conclusions are drawn.

II. Preliminary Concepts and Previous Work

Let $A = a_{n-1}a_{n-2}\cdots a_1a_0$ and $B = b_{n-1}b_{n-2}\cdots b_1b_0$ represent two binary numbers to be added and $S = s_{n-1}s_{n-2}\cdots s_1s_0$ be their sum. In the following, the symbols $\cdot$, $+$, $\oplus$, and $-$ are used to denote the AND, INCLUSIVE OR, EXCLUSIVE OR, and NOT logical operations, respectively. In binary addition, the computation of the carry signals is based on the following recursive formula:

$$c_i = g_i + z_i \cdot c_{i-1} \tag{1}$$

where $g_i = a_i \cdot b_i$ and z_i are the carry generate and the carry propagate terms, respectively. The latter, for the case of INCLUSIVE OR adders, is defined as $z_i = t_i = a_i + b_i$, while for the case of EXCLUSIVE OR adders, it is defined as $z_i = p_i = a_i \oplus b_i$. In Fig. 1, the implementation of the generate and the two types of propagate signals in domino CMOS logic is shown.

Expanding relation (1), each carry bit c_i can be expressed as

$$c_i = g_i + z_i g_{i-1} + z_i z_{i-1} g_{i-2} + \cdots + z_i z_{i-1}\cdots z_1 g_0 + z_i z_{i-1}\cdots z_0 c_{-1}. \tag{2}$$

The sum bits of the adder are defined as $s_i = p_i \oplus c_{i-1}$, where c_{-1} is the input carry.

The MCC [6], [7] generates all the carries computed according to relation (2) in parallel, using an iterative shared transistor structure. In practice, the CLA length is limited to four in order to cut down the number of series-connected transistors. Fig. 2 shows the conventional implementation of the 4-bit carry chain using multioutput domino CMOS logic.

MCC adders are EXCLUSIVE OR adders, i.e., the carry propagate signal is defined as $z_i = p_i = a_i \oplus b_i$, to avoid false discharges produced at the output nodes of the carry chain due to higher OR–AND forms of multioutput gates.

For the implementation of the sum signals, the domino chain is terminated, and the sum bits of the MCC adder are

Manuscript received November 23, 2012; revised May 22, 2013; accepted July 21, 2013. Date of publication September 16, 2013; date of current version October 14, 2013. This work was supported in part by the European Union (European Social Fund) and in part by Greek national resources under the framework of the "Archimedes III: Funding of Research Groups in TEI of Athens" project of the "Education & Lifelong Learning" Operational Program. This brief was recommended by Associate Editor Y. Miyanaga.

C. Efstathiou is with the Technological and Educational Institute of Athens, 12210 Athens, Greece (e-mail: cefsta@teiath.gr).

Z. Owda was with the University of Ioannina, 45110 Ioannina, Greece. He is now with the University of Siegen, 57076 Siegen, Germany (e-mail: zaher.owda@uni-siegen.com).

Y. Tsiatouhas is with the University of Ioannina, 45110 Ioannina, Greece (e-mail: tsiatouhas@cs.uoi.gr).

Color versions of one or more of the figures in this brief are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCSII.2013.2278088

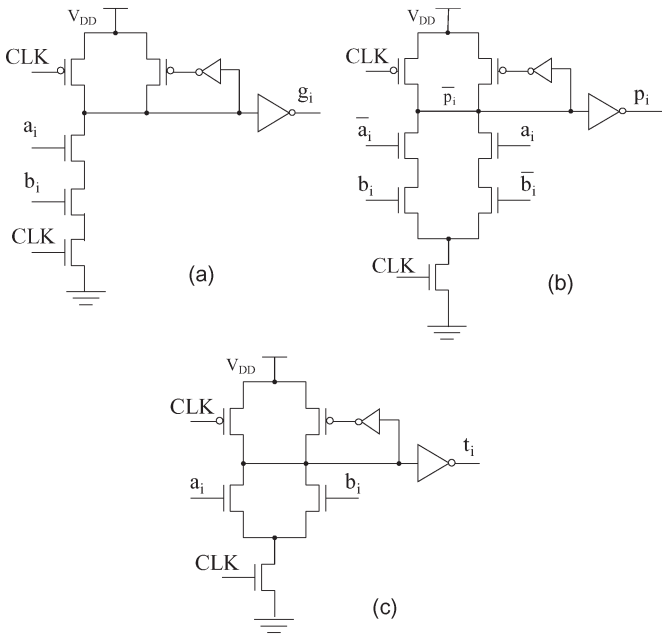

Fig. 1. Domino implementation for the (a) generate, (b) XOR propagate, and (c) OR propagate signals.

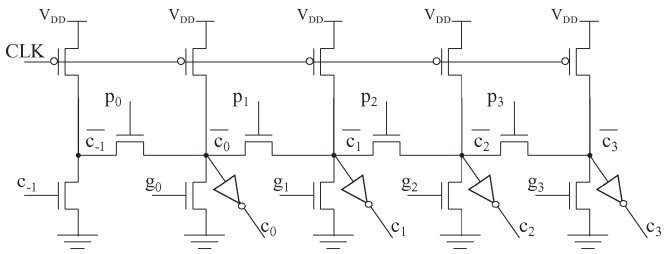

Fig. 2. Conventional domino 4-bit MCC.

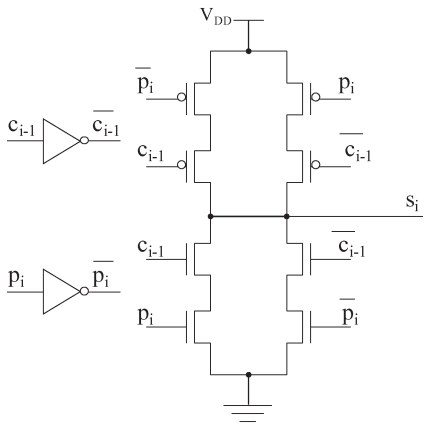

Fig. 3. Static CMOS implementation of the XOR gate for the sum computation.

implemented using static CMOS XOR gates [6], the design of which is shown in Fig. 3.

Several variations of the MCC adder in domino CMOS logic have been proposed in the literature [6]–[12]. Moreover, static CMOS MCC implementations are also given [13], [14]. Among them, a high-speed design has been proposed in [12], where the MCC is supported by the carry-skip capability to improve performance.

III. New High-Speed Double Carry Chain 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], [7], [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 [15]–[17]. 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.

A. 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

$$\begin{aligned} c_2 &= t_2(g_2 + g_1 + p_2 p_1 g_0 + p_2 p_1 p_0 c_{-1}) \\ &= t_2\left(g_2 + g_1 + p_2 p_1 t_0 (g_0 + c_{-1})\right) = t_2 \cdot h_2 \end{aligned}$$

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

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

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

B. Odd Carry Computation

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

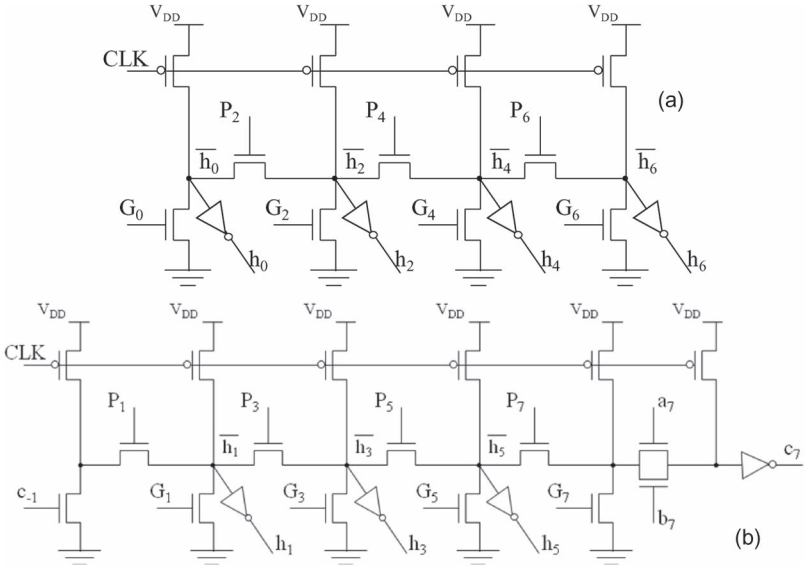

Fig. 4. Proposed carries' implementation for (a) the even carry chain and (b) the odd carry chain.

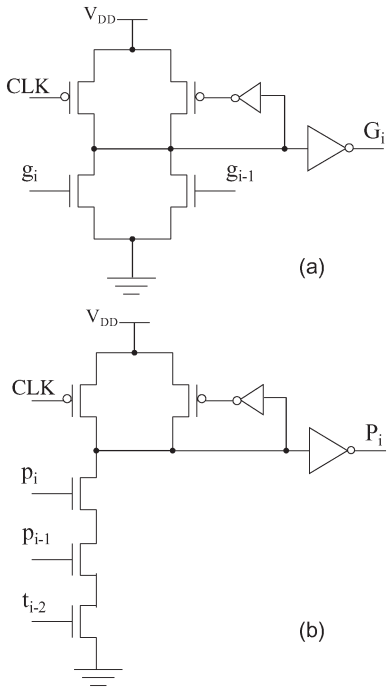

Fig. 5. New (a) generate and (b) propagate signals implemented in domino CMOS logic.

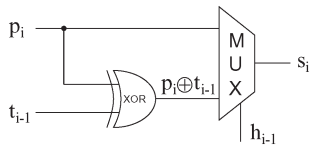

Fig. 6. Sum bit implementation.

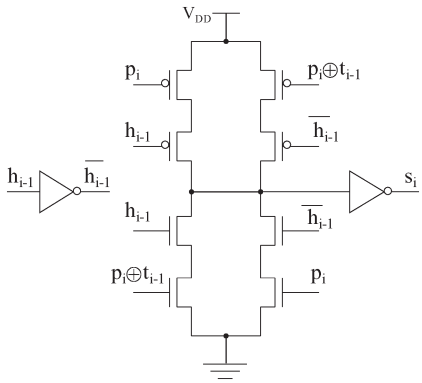

Fig. 7. Static CMOS implementation of the $2 \rightarrow 1$ multiplexer.

carries as follows:

$$h_1 = g_1 + g_0 + p_1 p_0 c_{-1}$$

$$h_3 = g_3 + g_2 + p_3 p_2 t_1 (g_1 + g_0 + p_1 p_0 c_{-1})$$

$$h_5 = g_5 + g_4 + p_5 p_4 t_3 (g_3 + g_2 + p_3 p_2 t_1 (g_1 + g_0 + p_1 p_0 c_{-1}))$$

$$h_7 = g_7 + g_6 + p_7 p_6 t_4 \times (g_5 + g_4 + p_5 p_4 t_3 \times (g_3 + g_2 + p_3 p_2 t_1 (g_1 + g_0 + p_1 p_0 c_{-1}))).$$

Let $G_i = g_i + g_{i-1}$ and $P_i = p_i \cdot p_{i-1} \cdot t_{i-2}$ be the new generate and propagate signals, respectively, where $g_{-1} = c_{-1}$ and $t_{-1} = 1$. Then, the following equations are derived for the new carries for even values of i:

$$h_2 = G_2 + P_2 G_0$$

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

$$h_6 = G_6 + P_6 G_4 + P_6 P_4 G_2 + P_6 P_4 P_2 G_0$$

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

$$h_1 = G_1 + P_1 c_{-1}$$

$$h_3 = G_3 + P_3 G_1 + P_3 P_1 c_{-1}$$

$$h_5 = G_5 + P_5 G_3 + P_5 P_3 G_1 + P_5 P_3 P_1 c_{-1}$$

$$h_7 = G_7 + P_7 G_5 + P_7 P_5 G_3 + P_7 P_5 P_3 G_1 + P_7 P_5 P_3 P_1 c_{-1}.$$

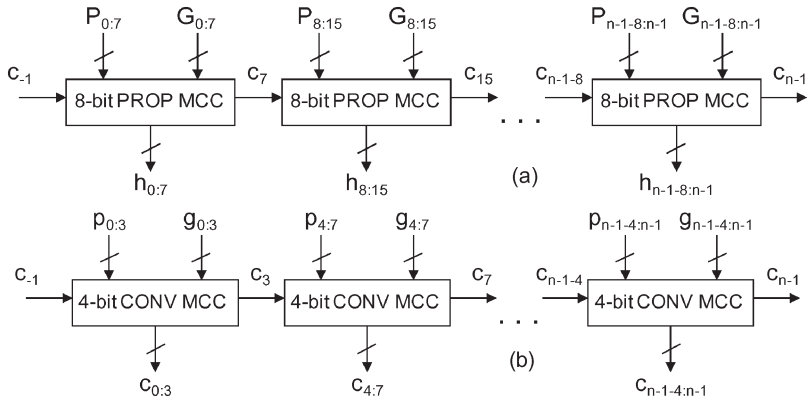

Fig. 8. Ripple carry chains based on (a) the proposed 8-bit MCC adder module and (b) the conventional 4-bit MCC adder module.

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, as shown in Fig. 4.

The new generate and propagate signals G_i and P_i can be easily proven to be mutually exclusive, avoiding false node discharges. Their domino CMOS implementation is shown in Fig. 5.

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 [17] and [18], the computation of the sum bits can be performed as follows:

$$s_i = \overline{h_{i-1}} \cdot p_i + h_{i-1} \cdot (p_i \oplus t_{i-1}) \tag{3}$$

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

Relation (3) 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}, as shown in Fig. 6.

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 (3).

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. The design of the XOR gate in Fig. 6 is similar to that in Fig. 3. An efficient static CMOS implementation of the $2 \rightarrow 1$ multiplexer is shown in Fig. 7.

IV. MCC Design Issues and Comparisons

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 carry chain principle given in Fig. 8(a) and (b), respectively, and simulated using SPECTRE in a standard 90-nm CMOS technology ($V_{\rm DD} = 1$ 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.

TABLE I
Carry Propagation Delay Times and Percentage Improvements

	PROP (ps)	CONV (ps)	Percentage (%)
8-bit	215.50	226.21	4.73
16-bit	352.40	458.15	23.08
32-bit	616.92	881.93	30.05
64-bit	1115.44	1718.20	35.08

The simulation results, for the carry propagation delays, are presented in Table I. The PROP design provides a performance improvement of 4.73% over the CONV design for the 8-bit adder. The performance improvements of the PROP design over the CONV design are 23.08% for the 16-bit adder, 30.05% for the 32-bit adder, and 35.08% for the 64-bit adder. Simulated waveforms of the carry signals (C_7, C_{15}, C_{23}, and C_{31}) for the proposed 32-bit adder are presented in Fig. 9. In all cases previously mentioned, the average energy consumption for a computation is increased by 43.4% for the PROP design with respect to the CONV one, while the area overhead is 49.9%, due to the extra gates that are required for the implementation of the t_i and the new generate (G_i) and propagate (P_i) signals. The proposed technique can be exploited in the design of arithmetic circuits where high performance is required at the expense of power consumption.

As referred previously, a modified high-speed design of the 4-bit MCC adder module has been proposed in [12], where the MCC is supported by the carry-skip capability to improve performance. The same technique can also be applied to the chain which computes the odd carries of the proposed 8-bit adder to further improve its efficiency. Since the 8-bit adder is the building block for higher bit adders, in all cases, the performance is proportionally improved. However, even without this addition, the proposed topology outperforms the modified MCC design proposed in [12] since it provides 7.18%, 11.79%, 17.59%, and 23.04% speed improvements for the 8-, 16-, 32-, and 64-bit adders, respectively, as it is presented in Table II. Moreover, in Fig. 10, a graphical presentation of the carry propagation delays with respect to the number of bits in the adder among the three design styles (PROP, CONV, and [12]) is given.

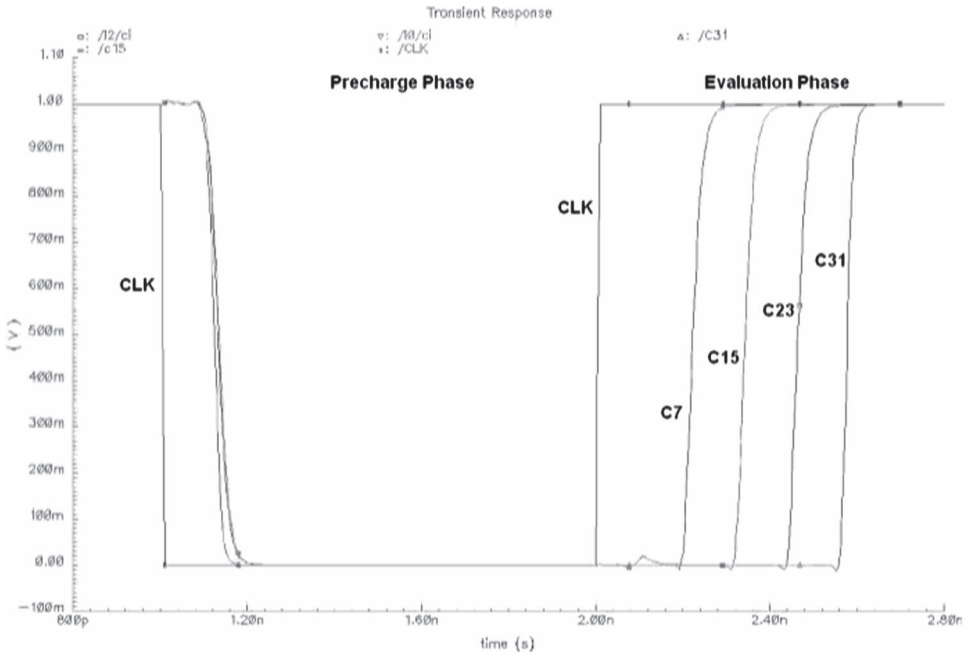

Fig. 9. Carry signal simulated waveforms for the proposed 32-bit adder.

TABLE II
CARRY PROPAGATION TIMES AND PERCENTAGE IMPROVEMENTS

	PROP (ps)	[12] (ps)	Percentage (%)
8-bit	215.50	232.18	7.18
16-bit	352.40	399.51	11.79
32-bit	616.92	748.60	17.59
64-bit	1115.44	1449.36	23.04

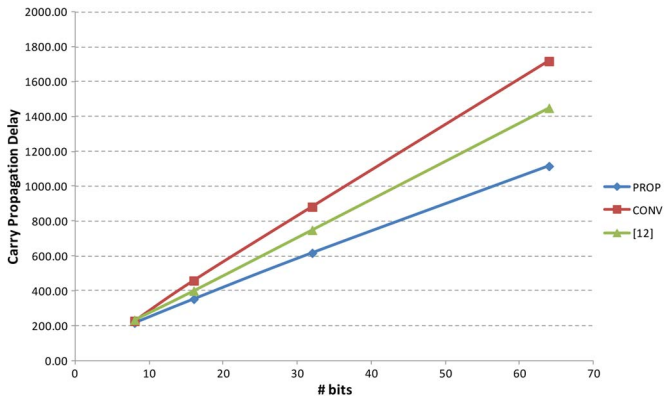

Fig. 10. Carry propagation delay with respect to the number of bits.

V. CONCLUSION

The MCC is an efficient and widely accepted design approach to construct CLA adders. In this brief, we have presented a new Manchester design style that is based on two independent carry chains. Each chain computes, in parallel with the other, half of the carries. In this way, the speed performance is significantly improved with respect to that of the standard MCC topology. The proposed design technique has been applied for the implementation of 8-, 16-, 32-, and 64-bit adders in multioutput domino logic, and the simulation results verified its efficiency.

REFERENCES

[1] K. Hwang, *Computer Arithmetic: Principles, Architecture, and Design*. New York, NY, USA: Wiley, 1979.
[2] B. Parhami, *Computer Arithmetic, Algorithms and Hardware*. New York, NY, USA: Oxford Univ. Press, 2000.
[3] I. Koren and A. K. Peters, *Computer Arithmetic Algorithms,* 2nd ed. Boca Raton, FL, USA: CRC Press, 2002.
[4] M. D. Ercegovac and T. Lang, *Digital Arithmetic*. San Mateo, CA, USA: Morgan Kaufmann, 2004.
[5] A. Weinberger and J. L. Smith, "A logic for high speed addition," *Nat. Bureau Stand. Circulation*, vol. 591, pp. 3–12, 1958.
[6] J. P. Uyemura, *CMOS Circuit Design*. Boston, MA, USA: Kluwer, 2001.
[7] N. Weste and D. Harris, *CMOS VLSI Design, A Circuit and System Perspective*. Reading, MA, USA: Addison-Wesley, 2004.
[8] P. K. Chan and M. D. F. Schlag, "Analysis and design of CMOS Manchester adders with variable carry-skip," *IEEE Trans. Comput.*, vol. 39, no. 8, pp. 983–992, Aug. 1990.
[9] Z. Wang, G. Jullien, W. Miller, J. Wang, and S. Bizzan, "Fast adders using enhanced multiple-output domino logic," *IEEE J. Solid State Circuits*, vol. 32, no. 2, pp. 206–214, Feb. 1997.
[10] S. Perri, P. Corsonello, F. Pezzimenti, and V. Kantabutra, "Fast and energy-efficient Manchester carry-bypass adders," *Proc. Inst. Elect. Eng.—Circuits Devices Syst.*, vol. 151, no. 6, pp. 497–502, Dec. 2004.
[11] M. Osorio, C. Sampaio, A. Reis, and R. Ribas, "Enhanced 32-bit carry look-ahead adder using multiple output enable-disable CMOS differential logic," in *Proc. 17th Symp. Integr. Circuits Syst. Design*, 2004, pp. 181–185.
[12] A. A. Amin, "Area-efficient high-speed carry chain," *Electron. Lett.*, vol. 43, no. 23, pp. 1258–1260, Nov. 2007.
[13] G. A. Ruiz, "New static multi-output carry look-ahead CMOS adders," *Proc. Inst. Elect. Eng.—Circuits, Devices Syst.*, vol. 144, no. 6, pp. 350–354, Dec. 1997.
[14] G. A. Ruiz and M. Granda, "An area-efficient static CMOS carry-select adder based on a compact carry look-ahead unit," *Microelectron. J.*, vol. 35, no. 12, pp. 939–944, Dec. 2004.
[15] H. Ling, "High-speed binary adder," *IBM J. Res. Develop.*, vol. 25, pp. 156–166, Mar. 1981.
[16] C. Efstathiou, H. T. Vergos, and D. Nikolos, "Ling adders in CMOS standard cell technologies," in *Proc. 9th ICECS*, Sep. 2002, vol. 2, pp. 485–489.
[17] G. Dimitrakopoulos and D. Nikolos, "High-speed parallel-prefix VLSI Ling adders," *IEEE Trans. Comput.*, vol. 54, no. 2, pp. 225–231, Feb. 2005.
[18] S. Vassiliadis, "Recursive equations for hardwired binary adders," *Int. J. Electron.*, vol. 67, no. 2, pp. 201–213, Aug. 1989.