Design and Implementation of Majority Logic of Parallel Adders Using Kogge-Stone Adder for Reducing Circuit Complexity Using FPGA

MAH JABBEN[1], V. PRADEEP KUMAR[2]
[1]PG Scholar, Dept of VLSI System Design, Bharat Institute Of Engineering And Technology, India,
[2]Assistant Professor, Dept of VLSI System Design, Bharat Institute Of Engineering And Technology, India,

Abstract: The design of high-performance adders has experienced a renewed interest in the last few years; among high performance schemes, parallel prefix adders constitute an important class. They require a logarithmic number of stages and are typically realized using AND-OR logic; moreover with the emergence of new device technologies based on majority logic, new and improved adder designs are possible. However, the best existing majority gate-based prefix adder incurs a delay of $2\log_2(n)-1$ (due to the nth carry); this is only marginally better than a design using only AND-OR gates (the latter design has a $2\log_2(n)+1$ gate delay). This paper initially shows that this delay is caused by the output carry equation in majority gate-based adders that is still largely defined in terms of AND-OR gates. In this paper, two new majority gate-based recursive techniques are proposed. The first technique relies on a novel formulation of the majority gate-based equations in the used group generate and group propagate hardware; this results in a new definition for the output carry, thus reducing the delay. The second contribution of this manuscript utilizes recursive properties of majority gates (through a novel operator)to reduce the circuit complexity of prefix adder designs. Overall, the proposed techniques result in the calculation of the output carry of an n-bit adder with only a majority gate delay of $\log_2(n)+1$. This leads to a reduction of 40percent in delay and30percent in circuit complexity (in terms of the number of majority gates) for multi-bit addition in comparison to the best existing designs found in the technical literature.

Keywords*:* Adder, Carry, Majority Voting Logic, Emerging Technologies, Arithmetic Complexity.

I. INTRODUCTION

The design of high performance multi-bit adders has been an active research topic for many years; schemes have been developed with the predominant goal of reducing the worst-case delay under a possible CMOS implementation. Existing multi-bit adders reduce the worst case delay based on strategies such as (i) unrolling the carry recurrence, and (ii) calculating the results prior to each possible carry input. Based on these strategies, several high performance designs have been proposed in the literature [1]. Among them, prefix adders constitute an important class, because they yield high performance at a relatively small fan-out and hardware requirements [2], [3], [4]. Prefix adders have been extensively used specially on critical paths [5] due to a compact yet fast implementation. Traditional prefix addition (as well as other schemes, such as carry look ahead) are based on generate and propagate signals derived using AND-OR logic [1]. An n-bit prefix adder requires only Oðlog2nÞ stages for the calculation of the delay. In terms of AND-OR gates, the Kogge-Stone and Ladner-Fischer adders incura delay of g $2\log_2(n)+1$ gates. Alternatives to AND-OR logic have also been considered for arithmetic circuit designs. In particular, majority logic has been of interest from the early 1960s. The realization of a one-bit adder using three majority gates and two inverters has been proposed as well as techniques based on decomposition and rearrangement to majority element-based synthesis of networks with limited fan-in components [6]. [7] has presented a geometric method that utilizes Veitch diagrams for synthesis using i-input majority gates for a variety of n-argument switching functions. An approach based on Logically Passive Self-Dual (LPSD) has been presented in [8]; an extension to this work has been presented in [9]. Interest in majority logic has been revived recently in the context of digital design for several emerging nanotechnologies (such as domain wall Nano-magnets[10], resonant tunneling diodes [11] and quantum dot cellular automata (QCA) [12]) [13], [14]. A few multi-bit adder designs in QCA have been reported [15], [16], [17], [18], [19]. However, the best existing majority gate-based n-bit adder design still incurs a delay of $2\log_2(n)-1$ for the nth carry, so only slightly better than using merely AND-OR gates. A close examination of this design reveals the cause for this limitation, i.e., the AND-OR logic has been primarily used to derive the majority gate-based designs. The goal of this paper is to first redefine the output carry of an n-bit adder in terms of only majority gates for delay reduction; the proposed formulations are useful for parallel adders not for synthesizing general majority gate circuits [13]. This paper provides two contributions to adder design using majority logic by which new majority gate-based recursive techniques are proposed. The first contribution is based on a new definition for the majority gate equations of the group generate and group propagate signals, such that the output carry is generated at a reduced delay. The second contribution is based on a new operator that exploits novel recursive properties of majority logic to achieve saving in circuit complexity (as given by the number of majority gates required in a design) for prefix adders. Overall, the proposed formulation results in calculating the output carry of an n-bit adder with a reduced majority gate delay of $\log_2(n)+1$. Moreover, the proposed approach leads to a reduction of 40percent in normalized delay and 30 percent in circuit complexity(in terms of required majority gates) for multi-bit addition compared to the best existing designs found in the technical literature.

II. MAJORITY LOGIC FORMULATION OF OUTPUT CARRY

A formulation for the carry output in terms of the Sum signal has been presented in the previous section; next the Sum signal is expressed in recursive majority logic form. This is used to efficiently generate the carry as in (6). Let $A = a_{n-1}\ a_{n-2} \ldots a_1 a_0$, $B = b_{n-1}\ b_{n-2} \ldots b_1 b_0$ and the (initial) carry C0 be inputs to an n-bit binary adder. Then, the output carry C_n can be defined in terms of majority gates as

$$C_n = M(a_{n-1}, b_{n-1}, M(a_{n-2}, b_{n-2}, \ldots, M(a_1, b_1, M(a_0, b_0, C_0))). \quad (1)$$

Hence, $Sum \geq 2^n$ and $Sum \geq (2^n - 1)$ are expressed in terms of majority gates using (7) and (8) as

$$Sum \geq 2^n = M(a_{n-1}, b_{n-1}, M(a_{n-2}, b_{n-2}, \ldots, M(a_1, b_1, M(a_0, b_0, 0)))) \quad (2)$$

$$Sum \geq (2^n - 1) = M(a_{n-1}, b_{n-1}, M(a_{n-2}, b_{n-2}, \ldots, M(a_1, b_1, M(a_0, b_0, 1)))) \quad (3)$$

Direct application of (9) to compute C_n is not efficient (i.e., it incurs in a high delay); so, the recursive forms of C_n, $Sum \geq 2^n$ and $Sum \geq (2^n - 1)$ in (1), (2), and (4) are needed to derive two of the contributions of this paper. For ease of notation, let $R_i(x)$ and $R_{i:j}(x)$ be defined as

$$R_i(x) = M(a_i, b_i, x) \quad (4)$$

$$R_{i:j}(x) = M(a_i, b_i, M(a_{i-1}, b_{i-1}, \ldots, M(a_{j-1}, b_{j-1}, M(a_j, b_j, x)))) \quad (5)$$

$$R_{i:i}(x) = R_i(x). \quad (6)$$

$Sum \geq 2^n$ and $Sum \geq (2^n - 1)$ in (10) and (11) can be rewritten using $R_{i:j}(x)$ as

$$Sum \geq 2^n = R_{n-1:1}(a_0 \cdot b_0) \quad (7)$$

$$Sum \geq (2^n - 1) = R_{n-1:1}(a_0 + b_0) \quad (8)$$

So the output carry C_n using (15) and (16) can be written as

$$C_n = M(R_{n-1:1}(a_0 \cdot b_0), R_{n-1:1}(a_0 + b_0), C_0). \quad (9)$$

(17) expresses the computation of all carries using C_0. For improvement in circuit complexity the lower order carries can be used for computation of the higher order carries. For example, when $n > i, C_n$ can be expressed using C_i as

$$C_n = M(a_{n-1}, b_{n-1}, M(a_{n-2}, b_{n-2}, \ldots, M(a_{i+1}, b_{i+1}, M(a_i, b_i, C_i)))). \quad (10)$$

TABLE I: Different Realizations for Carry Components in (5) Where $g_i = a_i\ b_i$, $k_i = a_i \otimes b_i$ and $p_i = (a_i + b_i)$

Realization	$Sum \geq 2^n$	$Sum = (2^n - 1)$
AND-OR Gates	$g_{n-1} + \sum_{i=n-2}^{0} (\prod_{j=n-1}^{i+1} k_j) g_i$	$\prod_{i=n-1}^{0} k_i$
Majority Gates [15]	$M(a_{n-1}, b_{n-1}, M(a_{n-2}, b_{n-2}, \ldots, M(a_1, b_1, M(a_0, b_0, 0))))$	$\prod_{i=n-1}^{0} p_i$
Majority Gates [14]	$M(a_{n-1}, b_{n-1}, M(a_{n-2}, b_{n-2}, \ldots, M(a_1, b_1, M(a_0, b_0, 0))))$	$\prod_{i=n-1, n-3, \ldots, 1} M(a_i, b_i, M(a_{i-1}, b_{i-1}, 1))$
Proposed	$M(a_{n-1}, b_{n-1}, M(a_{n-2}, b_{n-2}, \ldots, M(a_1, b_1, a_0)))$	$M(a_{n-1}, b_{n-1}, M(a_{n-2}, b_{n-2}, \ldots, M(a_1, b_1, b_0)))$

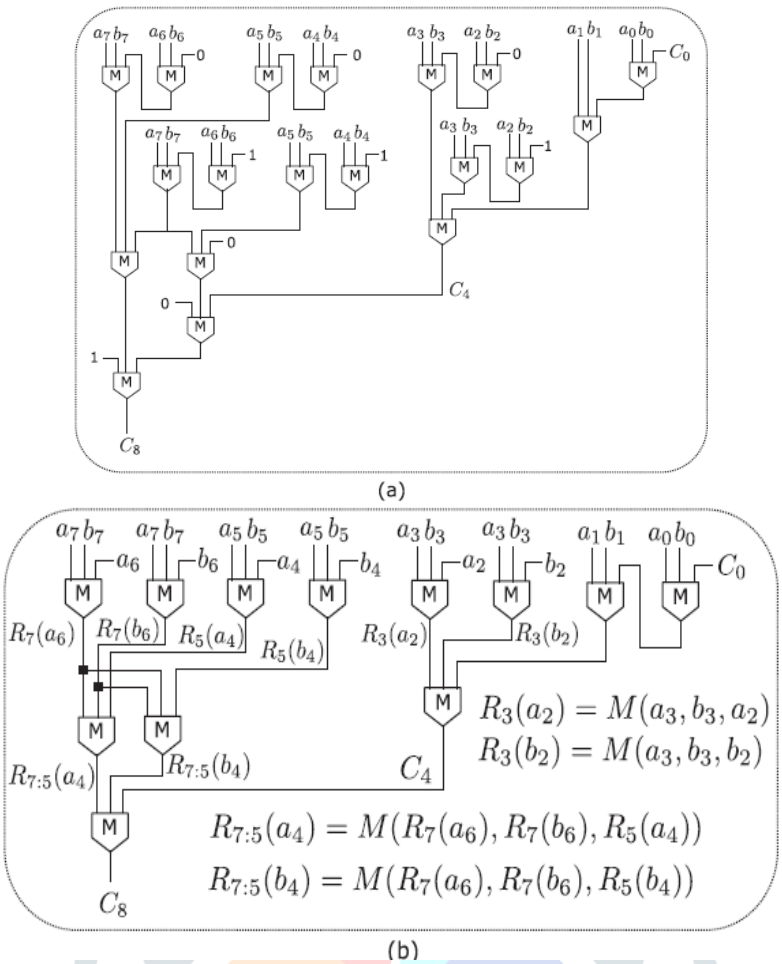

Fig.1. Carry C8 majority gate implementations. (a) Carry C8 of [14]. (b) Proposed majority gate diagram of carry C8.

Fig. 1b shows the proposed majority gate diagrams of C8using the proposed formulation. From Fig. 1, the proposed carryC4 requires 2 majority gates less than C4 of [17]. Similarly, C8 in Fig. 1 requires six majority gates less than in [17] (shown in Fig. 1), hence a saving of 33 percent is accomplished. The delay required for generating C8 is four $(\log_2(8)+1)$ majority gates (Fig.1) and this corresponds to a 20 percent saving compared to the design of [17]. Similarly, the delay required for generating C_n is $\log_2(n)+1$. The proposed majority gate formulation is applicable to various adders such as the Carry-Look Ahead adder (CLA), prefix adders, carry select adder and conditional sum adders. Due to space constraints, only the application to prefix adders is presented.

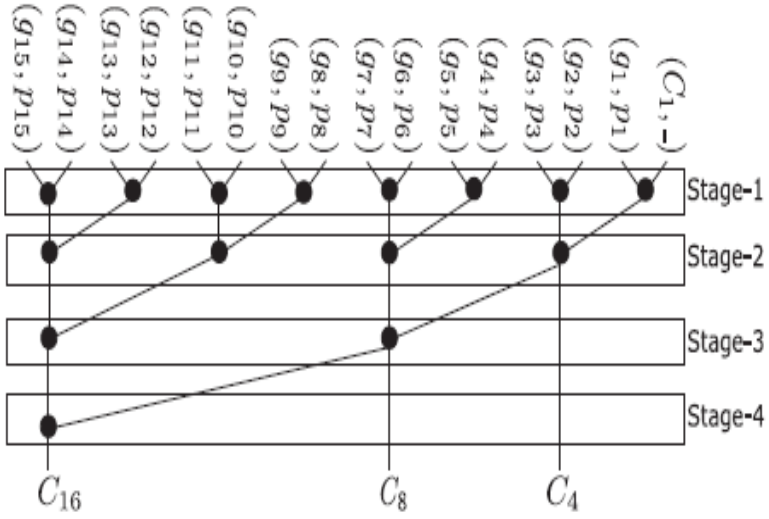

Fig.2. Conventional prefix graph of carry C_{16}. Where $g_i = a_i b_i$ and $p_i = a_i + b_i$.

III. PREFIX OPERATOR FOR ADDER DESIGN

Prefix adders are defined in terms of an associative operator [1] as in (30). The carry C4 is computed using the prefix associative operator in (31) such that generate $g_i = a_i\ b_i$ and *propagate* $p_i = a_i + b_i$.Fig. 2 shows its prefix graph. In this

section, a new associative operator is presented in terms of majority logic for obtaining a design with reduced complexity (in terms of number of gates and delay in the adder design)

$$(g_i, p_i) \circ (g_j, p_j) = (g_i + p_i p_j, p_i p_j) \quad \textbf{(11)}$$

$$(C_4, \_) = (g_3, p_3) \circ (g_2, p_2) \circ (g_1, p_1) \circ (C_1, \_). \quad \textbf{(12)}$$

The motivation for the definition of this new operator is as follows. The existing operator in (30) requires three majority gates;[17] has derived new properties to reduce the number of majority gates for such existing operator. However, this can be improved further, because it reduces the number of majority gates for only specific stages of the adder. In the scheme of [17] (Fig. 3) each prefix operation requires two majority gates each for stage-1 and stage-2,while the remaining stages require three majority gates each. The proposed design requires only two majority gates in all stages of prefix addition and completely eliminates the calculations of g_{is} and p_{is}. Fig.4 shows the proposed prefix operator symbol and the majority gate diagram. Fig. 5 shows the proposed prefix graph for calculating C16. The majority gate diagram of C16 is shown in Fig. 6 (the proposed prefix associative operator is marked with dotted linesFig.6) The proposed prefix operator has the following advantages compared to prior designs: (a) It completely removes the calculation of g_{is} and p_{is} from the prefix adders. (b) The computation of each prefix operator requires only two majority gates in all stages of prefix adders. (c) The computation of each prefix operator involves only one majority gate delay in all stages of the prefix adders. The proposed prefix operator is directly applicable to all types of prefix adder. In this paper, three types of prefix adders are considered, namely Kogge-Stone, Ladner-Fisher and Brent-Kung adders. The proposed majority gate diagrams for various prefix adders are shown in Fig. 7. Fig. 7a shows the proposed majority gate diagram of an 8-bit Kogge-Stone prefix adder; it requires three stages and four majority gates delay for calculation of all carries, so log2nstages for calculation of all carries in a n-bit adder. Fig. 7b shows the proposed majority gate diagram of an 8-bit Ladner-Fischer adder; it requires three stages and four majority gates delay for calculation of all carries. An n-bit Ladner-Fischer adder requires log2nstages for calculation of all carries. Fig. 7c shows the proposed majority gate diagram of an 8-bit Brent-Kung adder. It requires five stages and incurs in a six majority gates delay for the calculation of all carries. An n-bit Brent-Kung adder requires $2\log_2 n - 1$ stages for the calculation of the carries. The proposed prefix graphs of Kogge-Stone, Ladner-Fischer and Brent-Kung adders are similar to the conventional prefix-graphs [2], [3], [4], the significant difference is that the proposed prefix graphs do not require the calculation of the g_{is} and p_{is}.

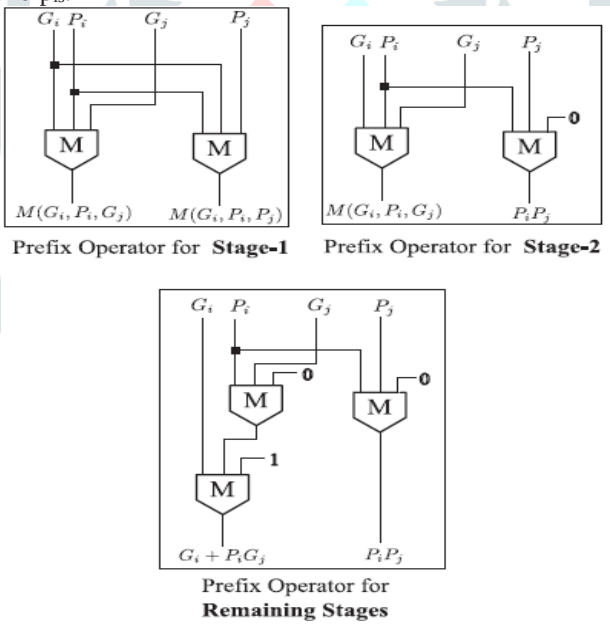

Fig.3. Prefix-Adder associative operator of [14].

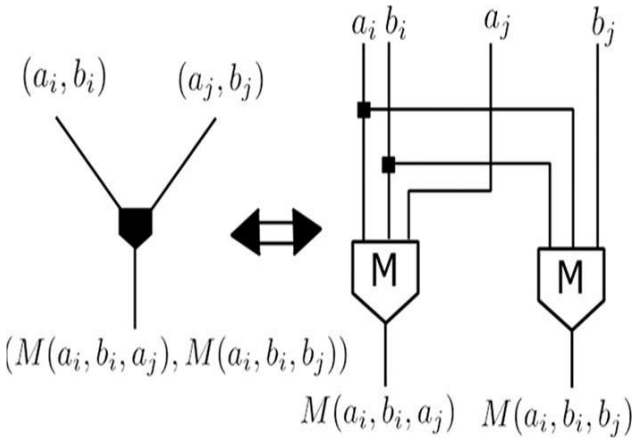

Fig.4. Proposed prefix operator symbol and majority gate diagram.

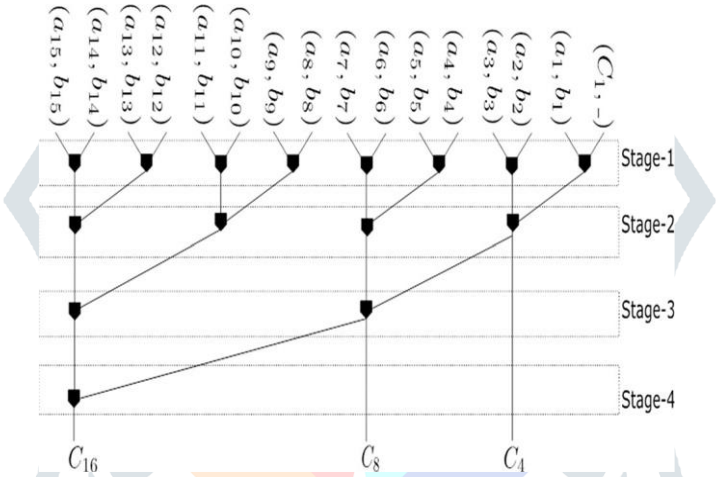

Fig.5. Proposed prefix graph of carry C16.

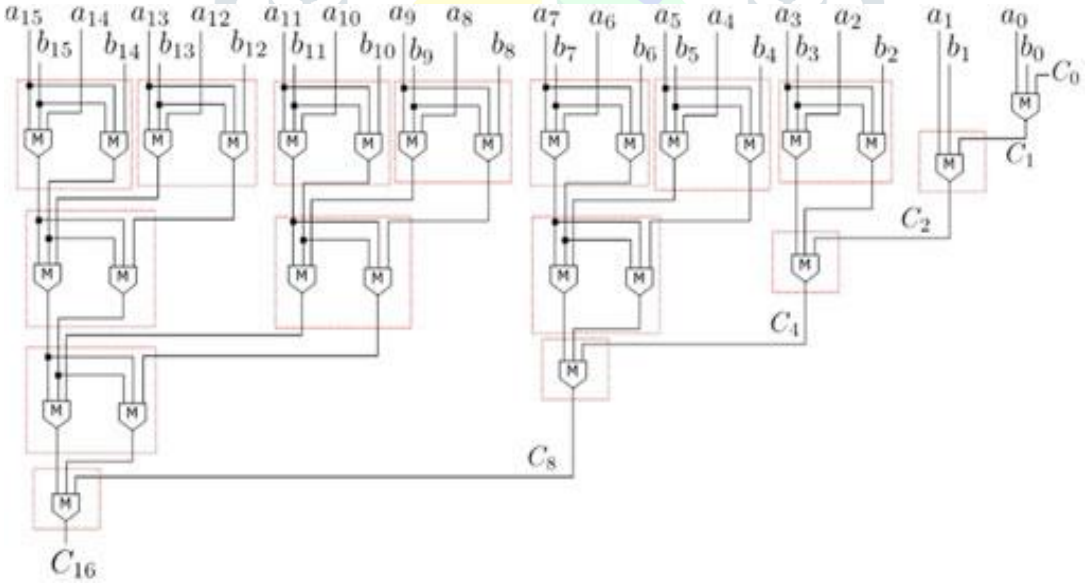

Fig.6. Majority gate diagram of carry C16 of proposed prefix adder (C16 is calculatedthe same way for various adder schemes such as Kogge-Stone, Ladner-Fischer and Brent-Kung).

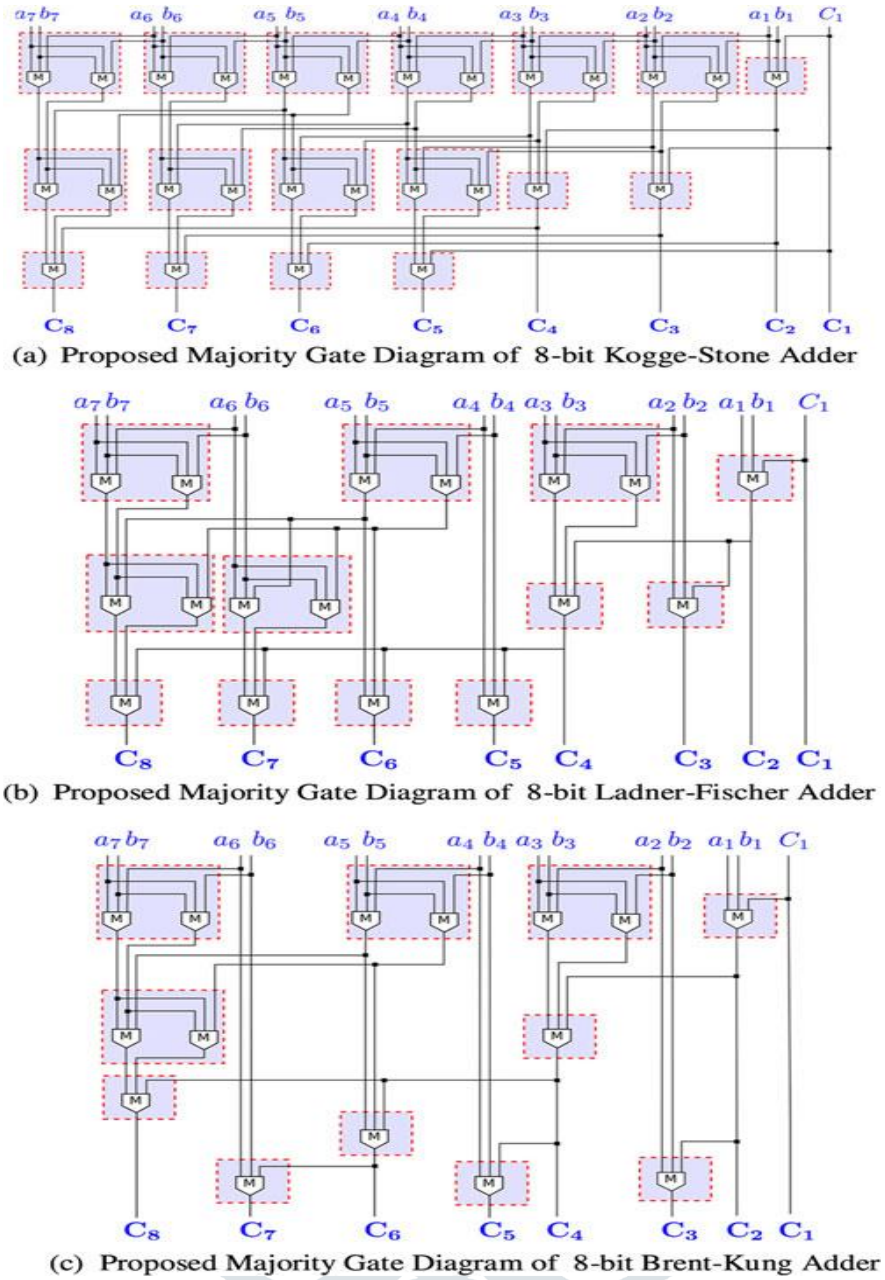

Fig.7. Majority gate diagrams of different 8-bit prefix adders using proposed prefix operator.

IV. SIMULATION RESULT

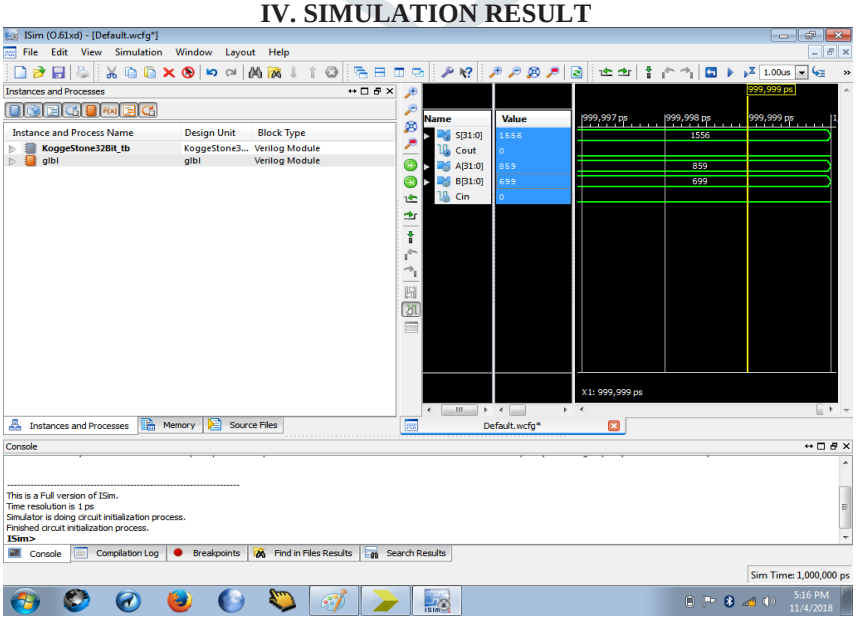

Fig.8.result.

V. CONCLUSION

A new majority gate-based approach for high performance adder design has been presented. The two contributions of this manuscript(the formulation of the carry output and the recursive prefix operator for majority logic) have resulted in a reduction in circuit complexity (as requiring a lower number of majority gates in an adder design) as well as lower propagation delay for the leading carry. The proposed strategy has been applied to various prefix adders including the Kogge-Stone, Ladner-Fischer and Brent-Kung adders. It is observed that these new results achieve a reduction in delay of at least log2n over the best existing majority gate-based adders found in the technical literature. Specifically, reductions of40 percent in delay and 30 percent in circuit complexity (in terms of the number of majority gates) has been accomplished for multi bit adder schemes. As shown in Table 2, $I_s(n)$ remains constant and for very large values of n, $I_c(n)$and the normalized delay $d(n)$show considerable reductions, for example a nearly 50 percent for $d(n)$for the three prefix adders considered in this manuscript. Current research deals with the applications of these findings to emerging technologies and in particular, the implications on different applications requiring fast arithmetic processing.

VI. REFERENCES

[1] I. Koren, Computer Arithmetic Algorithms. Natick, MA, USA: A.K. PetersLtd., 2002.
[2] P. Kogge and H. Stone, "A parallel algorithm for the efficient solution of ageneral class of recurrent equations," IEEE Trans. Comput., vol. C-22, no. 8,pp. 786–793, Aug. 1973.
[3] R. Ladner and M. Fischer, "Parallel prefix computation," J. Assoc. Comput.Mach., vol. 27, no. 4, pp. 831–838, Oct. 1980.
[4] R. P. Brent and H. T. Kung, "A regular layout for parallel adders," IEEETrans. Comput., vol. C-31, no. 3, pp. 260–264, Mar. 1982.
[5] D. Harris and S. Harris, Digital Design and Computer Architecture, 2nd ed.Burlington, MA, USA: Morgan Kaufmann, 2012.
[6] F. Miyata, "Realization of arbitrary logical functions using majorityelements," IEEE Trans. Electron.Comput., vol. EC-12, no. 3, pp. 183–191,Jun. 1963.
[7] H. S.Miller and R. O.Winder, "Majority-logic synthesis by geometric methods,"IEEE Trans. Electron.Comput., vol. EC-11, no. 1, pp. 89–90, Feb. 1962.
[8] S. B. Akers, "Synthesis of combinational logic using three-input majoritygates," in Proc. 3rd Annu. Symp. Switching Circuit Theory Logical Des., 1962,pp. 149–185.
[9] E. Riseman, "A realization algorithm using three-input majority elements,"IEEE Trans. Electron.Comput., vol. EC-16, no. 4, pp. 456–462, Sep. 1967.
[10] A. Roohi, R. Zand, and R. F. DeMara, "A tunable majority gate-based fulladder using current-induced domain wall nanomagnets," IEEE Trans.Magn., vol. 52, no. 8, pp. 1–7, Aug. 2016.
[11] P. Mazumder, S. Kulkarni, M. Bhattacharya, J. Sun, and G. Haddad,"Digital circuit applications of resonant tunneling devices," Proc. IEEE,vol. 86, no. 4, pp. 664–686, Aug. 1998.
[12] P. Tougaw and C. Lent, "Logical devices implemented using quantumcellular automata," J. Appl. Phys., vol. 75, no. 3, pp. 1818–1825, 1994.
[13] R. Zhang, P. Gupta, and N. K. Jha, "Majority and minority network synthesiswith application to QCA-, SET-, and TPL-based nanotechnologies,"IEEE Trans. Comput.-Aided Des.Integr. Circuits Syst., vol. 26, no. 7, pp. 1233–
1245, Jul. 2007.
[14] L. Amaru, P. E. Gaillardon, A. Chattopadhyay, and G. De Micheli, "Asound and complete axiomatization of majority- n logic," IEEE Trans. Comput.,vol. 65, no. 9, pp. 2889–2895, Sep. 2016.
[15] H. Cho and E. Swartzlander, "Adder and multiplier designs in quantumdotcellular automata," IEEE Trans. Comput., vol. 58, no. 6, pp. 721–727,Jun. 2009.
[16] V. Pudi and K. Sridharan, "Low complexity design of ripple carry andbrent-kung adders in QCA," IEEE Trans. Nanotechnol., vol. 11, no. 1,pp. 105–119, Jan. 2012.
[17] S. Perri and P. Corsonello, "New methodology for the design of efficientbinary addition circuits in QCA," IEEE Trans. Nanotechnol., vol. 11, no. 6,pp. 1192–1200, Nov. 2012.
[18] V. Pudi and K. Sridharan, "New decomposition theorems on majority logicfor low-delay adder designs in quantum dot cellular automata," IEEETrans.Circuits Syst. II: Express Briefs, vol. 59, no. 10, pp. 678–682, Oct. 2012.
[19] S. Perri, P. Corsonello, and G. Cocorullo, "Area-delay efficient binary addersin QCA," IEEE Trans. Very Large Scale Integr. VLSI Syst., vol. 22, no. 5,pp. 1174–1179, May 2014.
[20] R. Lindaman, "A theorem for deriving majority-logic networks within anaugmented Boolean algebra," IEEE Trans. Electron.Comput., vol. EC-9,no. 3, pp. 338–342, Sep. 1960.

Author's Details:

MAH JABEEN (STUDENT) Dept of Vlsi System Design, Bharat Institute Of Engineering And Technology, EMAILID: mahjabeen2761994@gmail.com.

Prof.G.KumaraswamyRao, Director(R&D),BIET(former Director & O.S. of DRDL Ministry of Defence) Prof.G.Kumaraswamy, former director of Defence Electronics Research Lab an Electronics Systems Engineering center, was associated in work with DR.A.P.J.abdulkalam in the "Integrated Guided Missile Program, and was personally responsible for the design and development of a number of missile guidance electronic systems for Trishul, Prithvi and Akash missiles". He has also designed the ground guidance electronic system for india's first short range surface to air missile. Prof.G. kumaraswamy is the director of R& D at Bharat Institute of Engineering and Technology.

V PRADEEP KUMAR, Dept of VLSI system design, as assistant professor 9 yrs in teaching and 2- yrs in industry,biet4ru@gmail.com.