Binary Adder Using More Efficient Area And Time Optimized Quantum-Dot Cellular Automata

G.Jhansi [1], Gera Indira PriyaDarshini [2]
[1]Assistant Professor, [2]Associate Professor
Department of Electronics & Communication Engineering, NNRES Group of Institutions, India

Abstract - The area and complexity are the major issues in circuit design. The quantum-dot cellular automata (QCA) approach represents one of the possible solutions in overcoming the physical limit of reduction of transistor size to increase the computational capabilities, even though the design of logic modules in QCA is not always straightforward. The quantum dot cellular automata can implement digital circuits with faster speed, smaller size and low power consumption. Adders are the logic circuits designed to perform high speed arithmetic operations and are important components in digital systems because of their extensive use in other basic operations such as subtraction, multiplication and division. A new adder designed in QCA was presented. It achieved speed performances higher than all the existing QCA adders, with an area requirement comparable with the cheap RCA and CFA demonstrated. The novel adder operated in the RCA fashion, but it could propagate a carry signal through a number of cascaded MGs significantly lower than conventional RCA adders.

Keywords - QCA, Adder, Area, Major Gates, Efficiency

I. INTRODUCTION

In electronics, an adder or summer is a digital circuit that performs addition of numbers. In many computers and other kinds of processors, adders are used not only in the arithmetic logic unit(s), but also in other parts of the processor, where they are used to calculate addresses, table indices, and similar operations. Although adders can be constructed for many numerical representations, such as binary-coded decimal or excess-3, the most common adders operate on binary numbers. In cases where two's complement or ones' complement is being used to represent negative numbers, it is trivial to modify an adder into an adder–Subtractor. Other signed number representations require a more complex adder. To reduce the computation time, engineers devised faster ways to add two binary numbers by using carry-look ahead adders. They work by creating two signals (P and G) for each bit position, based on whether a carry is propagated through from a least significant bit position (at least one input is a '1'), generated in that bit position (both inputs are '1'), or killed in that bit position (both inputs are '0'). In most cases, P is simply the sum output of a half-adder and G is the carry output of the same adder. After P and G are generated the carries for every bit position are created. Some advanced carry-look ahead architectures are the Manchester carry chain, Brent–Kung adder and the Kogge–Stone adder.

Some other multi-bit adder architectures break the adder into blocks. It is possible to vary the length of these blocks based on the propagation delay of the circuits to optimize computation time. These block based adders include the carry-skip (or carry-bypass) adder which will determine P and G values for each block rather than each bit, and the carry select adder which pre-generates the sum and carry values for either possible carry input (0 or 1) to the block, using multiplexers to select the appropriate result when the carry bit is known.

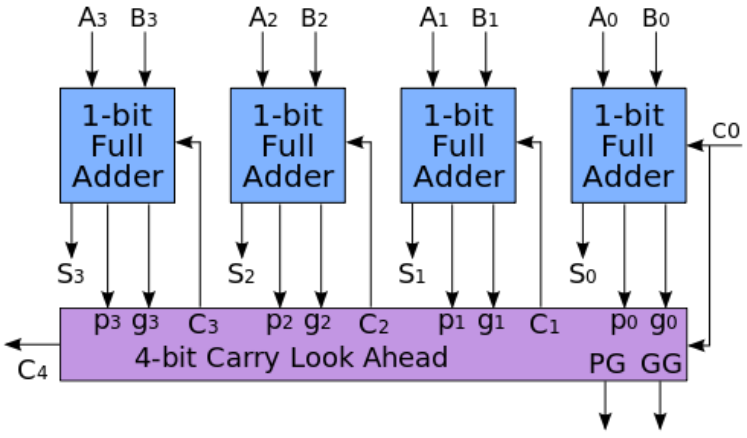

Fig: 1. Carry-Look A head Adder

Other adder designs include the carry-select adder, conditional sum adder, carry-skip adder, and carry-complete adder. By combining multiple carry look ahead adders even larger adders can be created. This can be used at multiple levels to make even larger adders. For example, the following adder is a 64-bit adder that uses four 16-bit CLAs with two levels of LCUs.

The carry look ahead adder (CLA) has a regular structure. It achieves high speed. In this paper we design a 4 bit CLA. This design employs 4 bit slices for the look ahead logic. These adders avoid feedback signals that are used in regular CMOS. By the nature of QCA cells the carry look ahead adder is pipelined. In the PG block we have a generated output that indicates that a carry is "generated "at bit position and a propagate output that indicates that a carry entering bit position will propagate to the next bit position. Thus these are used to produce all the carries in parallel at the successive blocks. Due to the pipeline diagram all sum signals are available at the same clock period. We represent QCA design of CLA.

II. PROPOSED SYSTEM

A quantum dot is a site in a cell in which a charge can be localized. The cell contains two extra mobile electrons that can quantum mechanically tunnel between dots, but not cells. In the ground state and in the absence of external electrostatic perturbation, the electrons are forced to the corner positions to maximize their separation due to Coulomb repulsion. The two possible charge configurations are used to represent binary "0" and "1". Note that in the case of an isolated cell, the two polarization states are energetically degenerate. However the presence of other charges (neighbour cells) breaks the degeneracy and one polarization state becomes the cell ground state. Majority gates are the simple and fast logic to transfer the input to output. In this logic the number of bits which are maximum at input is transferred at the output. For example if we take the example of three bit binary number 001 at the input that contains 2 zero's in the binary number then this 0 is transferred at the output.

Design which provides low power consumption implemented on FPGA. After design the final architecture is synthesized on various FPGA Architectures & comparison is done based on area and power consumption. In the previous papers implementation is done for 3bit majority logic. But in this particular paper the main focus is the implementation of 4bit majority logic on FPGA (Field Programmable Gate Array) used for design verification. From the available FPGA architectures design is implemented on the FPGA that provides the low chip area (lowest gate count) & low power consumption. So in the present architecture efforts are done to reduce the power and chip area. In the Digital IC design area, speed & power consumption are the parameters which are available for optimization. When an IC is designed the power is consumed to charge the parasitic capacitance of the interconnections. So by choosing the particular material this factor can be reduced by the significant amount decided by the process technology. So by choosing a particular FPGA package required for final implementation one can reduce the parasitic effects in the architecture design.

INPUT A	INPUT B	INPUT C	OUTPUT
0	0	0	0
1	0	0	0
0	1	0	0
0	0	1	0
1	1	0	1
1	0	1	1
0	1	1	1
1	1	1	1

Table 1. Truth table of a three-input majority logic gate

In QCA design, both logic structure and interconnections are made of QCA cells that can utilize the bridge technique. The fundamental logic gates of QCA technology are majority gate (MG) and inverter. The logic function of MG for three inputs A, B, C is reported in.

$$M(ABC) = A.B + B.C + A.C \quad (1)$$

The input cell belongs to the same clock signal. The three input majority gate is shown in Figure. 1.

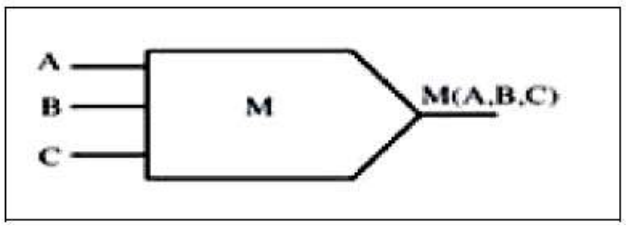

Fig:2. Three input Majority gate

The majority gate function as both AND and OR gate. When third input is 0 the MG act as AND gate and produces generate term and when third input is 1 the MG acts as OR gate and generates propagate term. The corresponding equations are reported in (2) and (3).

$$MG(A; B; C) = A.B \text{ when } C = 0 \quad (2)$$
$$MG(A; B; C) = A+B \text{ when } C = 1 \quad (3)$$

The majority gate is shown in Fig. 3 respectively. The majority gate performs a three-input logic function. Assuming the inputs are A, B and C, the logic function of the majority gate is

$$m(A, B,C) = A! B + B !C + A!C \quad (1)$$

By fixing the polarization of one input as logic "1" or "0", we can obtain an OR gate and an AND gate respectively. More complex logic circuits can then be constructed from OR and AND gates.

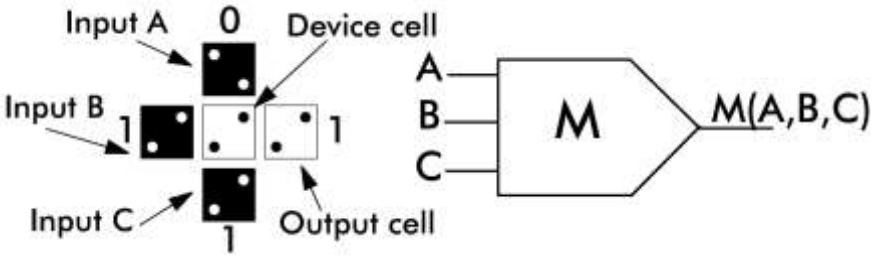

Fig:3. A QCA majority gate

To introduce the novel architecture proposed for implementing ripple adders in QCA, let consider two n-bit addends A = an−1, . . . , a0 and B = bn−1, . . . , b0 and suppose that for the I th bit position (with i = n − 1, . . . , 0) the auxiliary propagate and generate signals, namely pi = ai + bi and gi = ai · bi , are computed. ci being the carry produced at the generic (i−1) bit position, the carry signal ci+2, furnished at the (i+1)th bit position, can be computed using the conventional CLA logic reported in (2). In this way, the RCA action, needed to propagate the carry ci through the two subsequent bit positions, requires only one MG.

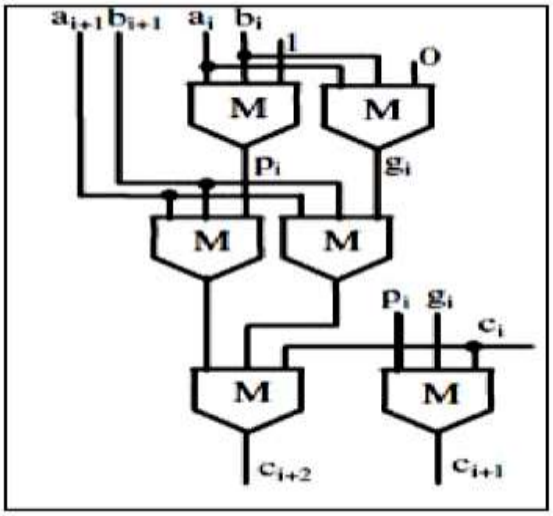

Fig:4. novel 2 –bit basic module

III. METHOD OF IMPLEMENTATION

One bit full adder has been implemented in two different methods. The first method was conventional (Direct implementation) and consumed a lot of hardware. The second method is (Majority gate reduction) simple and have less hardware requirement. Here we apply the Majority logic method for constructing QCA adders. The proposed adders are implemented with QCA cells and number of cells has reduced by cell minimization techniques. Hence our implementation further reduces the area and complexity.

The half adder is designed with 4 majority gates and 2 inverters as shown in Fig. 5.27. A one bit full adder circuit is constructed by the two half adder circuits and an OR gate. The full adder is designed with 9 majority gates and 4 inverters. QCA implementation of the half adder and the full adder. The total number of cells required to implement a half adder is 77, with an area of 83160 nm2 which is much lesser than the previous implementations. The previous implementation has 105 cells with an area of 108000nm2. The full adder is designed with 3 majority gates and 2 inverters.

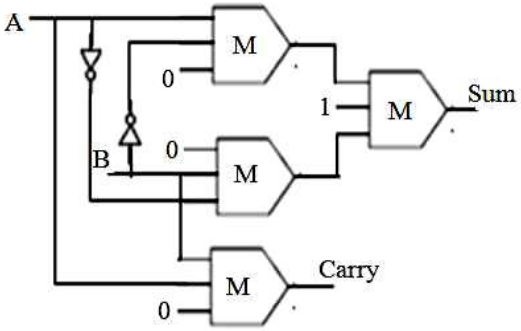

Fig5. Half adder schematic

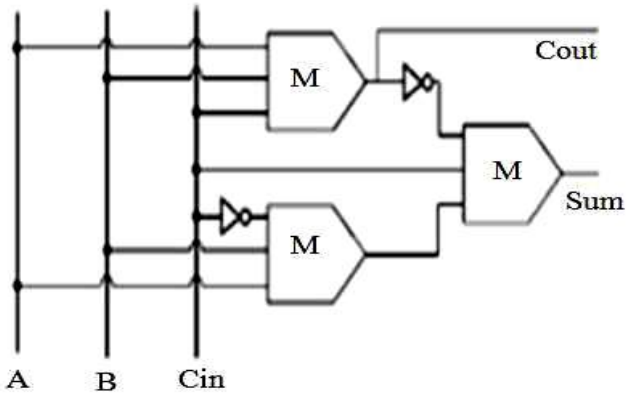

Fig6. Full adder schematic

Designers face many problems related to shrinking feature sizes of IC processes. Cross-talk, electro migration, self-heat, and other problems become more and more important. Cross-talk can lead to unpredictable design behaviours, high variation of delays and signal integrity problems in a wire depending on a state of a neighbour wire and thus should be avoided at any cost. It proposes to solve the problem of cross-talk-immune IC design by using predetermined layout patterns on the IC, called the "layout fabric". It can be observed that the same problems exist in new technologies such as Quantum Dot Cellular Automata.

Bridging together logic synthesis and layout synthesis proceeds in several directions. One approach to this problem, exemplified by, makes logic synthesis aware of the layout early in the design flow. Another approach aims at creating specific layout structures with regular properties and synthesizing circuits using these structures. Research in regular structures to implement Boolean functions has started with the work of Akers. The advantage of the regular layout fabrics is that they guarantee short wire length, predictable delay, and the absence of crosstalk. The disadvantage is that it may be difficult to derive compact representation for relatively complex logic functions.

There have been recently many papers published on Quantum Dots Cellular Automata (QCA). These papers use the Majority and Inverter gates for logic. Although the authors appreciate regularity in their designs of special functions, they did not consider general-purpose design methods to design arbitrary combinational logic blocks.

In this section a new QCA addition algorithm and the corresponding one-bit QCA adder structure that reduces the number of the majority gates and inverters required by existing designs. Then, we demonstrate that, using this structure, we can obtain efficient carry look ahead n-bit QCA adders.

The one-bit QCA adder consists of three majority gates and two inverters. It results in reduced hardware compared to the original full adder and retains the simple clocking scheme. It is noted that the bit-serial QCA adder uses a variant of the one-bit QCA adder. To create an n-bit adder, we arrange n number of one-bit adders vertically in a column. The clocking of the cells within the n-bit adder is designed such that the carry will propagate down to the last bit before the sum is calculated, thereby implementing a CLA adder.

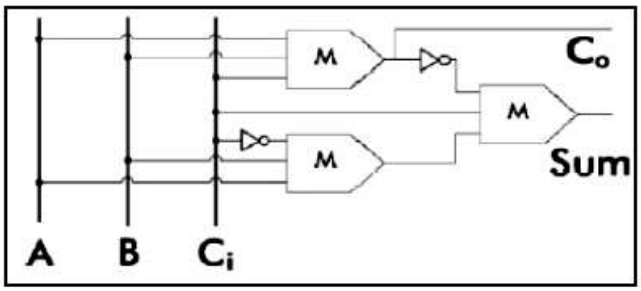

Fig:7. one-bit QCA Full Adder

The QCA enhancement adder design requires fewer majority gates and inverters while maintaining the same clocking scheme and speed in comparison with existing QCA adders. Compared to the bit-serial QCA adder, the QCA enhancement design is n times faster, at the expense of being approximately n times larger in area. The performance data of the QCA enhancement adder and of those considered in. It is seen that the QCA enhancement design requires less hardware while retaining the simple clocking scheme and parallel structure of the original carry look-ahead approach. a 45-degree wire at the proper location. Implementing the logical AND and OR functions is also quite simple.

The logical function for the majority gate is:

$$Y = AB + BC + AC$$

The AND function can be implemented by setting one value (A, B, or C) in equation to a logical 0. Similarly, the OR function can be implemented by Setting one value (A, B, or C) in equation to a logical 1This results in the equations:

$$AND = AB + B(0) + A(0) = AB$$
$$OR = AB + B(1) + A(1) = A + B$$

To overcome the drawbacks of the QCA system we have introduced a small change in QCA enhancement system. In QCA system we are used more majority gates to generate carry's and sum's. To generate a carry the inputs is given to the one majority gate and same inputs given to the other majority gate to generate the sum. By using two majority gates for same inputs the area and time delay increased. By using one majority gate instead of using two majority gates for the same inputs to generate carry and sum.

IV. SIMULATION RESULTS

In this QCA system the numbers of majority gates are high to implement 64 bit binary adder in QCA. By using this process the circuit complexity is very high and area is also maximized. Hence Time delay is also increased. To overcome this circuit complexity the numbers of majority gates are reduced so automatically area and time delay also decreases.

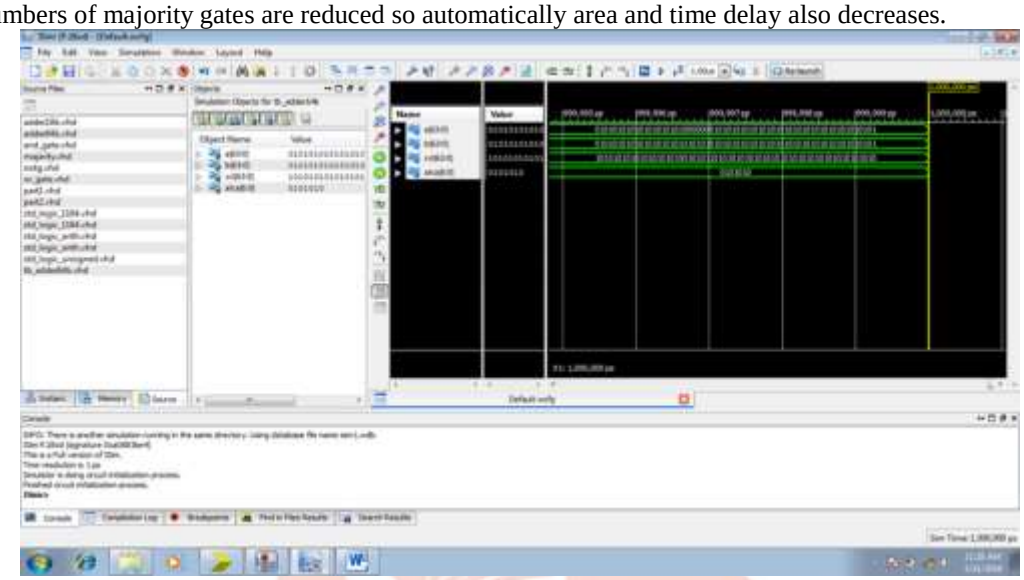

Fig:8. simulation result

V. CONCLUSION

A new adder designed in QCA was presented. It achieved speed performances higher than all the existing QCA adders, with an area requirement comparable with the cheap RCA and CFA demonstrated. The novel adder operated in the RCA fashion, but it could propagate a carry signal through a number of cascaded MGs significantly lower than conventional RCA adders. A 64-bit binary adder designed as described in this brief exhibited a delay and occupied an active area. It achieved speed performances higher than all the existing QCA adders, with an area requirement comparable.

VI. REFERENCES

[1] P. Farm and E. Dubrova, Technology Mapping for Chemically Assembled Electronic Nanotechnology, Proc. IWLS '02.
[2] S. Goldstein and M. Budiu, Nanofabrics: Spatial computing using molecular electronics, Proc. 28th Annual International Symposium on Computer Architecture, (Gothenborg, Sweden), June 2001.
[3] H. Hasegawa, A. Ito, Ch. Jiang, and T. Muranaka, Atomic Assisted Selective MBE Growth of InGaAs Linear and Hexagonal Nanowire Networks For Novel Quantum Circuits, Proc. 4th Intern Workshop on Novel Index Surfaces (NIS'01), Sept 16-20, Apset France.
[4] S.P. Khatri, R.K. Brayton, A.L. Sangiovanni-Vincentelli, Cross-Talk Noise Immune VLSI Design Using Regular Layout Fabrics, Kluwer Academic Publishers, Boston, Hardbound, ISBN 0-7923, 7407-X, June 2001, 144 pp.
[5] C. Lent, P. Tougaw, W. Porod, and G. Bernstein, Quantum cellular automata, Nanotechnology, 4: pp. 49-57, 1993.
[6] J. P. Marques-Silva, K. A. Sakallah, GRASP: A Search Algorithm for Propositional Satisfiability. IEEE Trans. Comp. Vol. 48, No. 5, May 1999, pp. 506-521.
[7] M. W. Moskewicz et al., Chaff: Engineering an Efficient SAT Solver. Proc. DAC'01, pp. 530-535.
[8] A. Mukherjee, R. Sudhakar, M. Marek-Sadowska, S. I. Long, Wave Steering in 466-471.
[9] M. T. Niemier, A. F. Rodriques, P.M. Kogge, A Potentially Implementable FPGA for Quantum Dot Cellular Automata, Workshop on Non-Silicon Computation (NSC), in conjunction with Int. Symp. On High Performance Computer Architecture (HPCA), February 3, 2003.
[10] M. Perkowski, and E. Pierzchala, New Canonical Forms for Four-Valued Logic, Report, Electrical Engineering Department, PSU . 1993.
Programmable Analog Array Circuit.