NEW DESIGN APPROACH TO IMPLEMENT BINARY ADDER BY USING QCA

K.Lavanya[1], S.Ramanjinaik[2]

[1] *M.Tech,VLSI Design, Student, SRIT Engg.College, karanamlavanya0482@gmail.com.*

[2] *Associate.Professor.in ECE, SRIT Engg.College,AP,India, ramanjivlsi@gmail.com.*

---***---

Abstract *: Now a day's certain limit has specified for the transistor count in IC's. Hence to incorporate more number of transistors in a single die to increase computational capabilities a new gate has been implemented in this paper that overcomes the physical limit of the existing designs. The new technique implemented is quantum dot cellular automata (QCA), which is the design of logic modules in QCA.A new 128-bit adder is implemented which will be more efficient of delay and area. The 128-bit adder implemented gives a delay of 18.77 ns and number of LUTS 129.*

***Key Words*: Adders, nanocomputing, quantum-dot cellular automata(QCA), Xilinx 13.1i, Ripple carry adder (RCA).**

1.INTRODUCTION

Quantum-dot Cellular Automata (QCA) is a new nano computing paradigm which encodes binary information by charge configuration within a cell instead of the conventional current switches. There is no current flow within the cells since the columbic interaction between the electrons is sufficient for computation. This paradigm provides one of many possible solutions for transistor-less computation at the nano scale.

The standard QCA cells have four quantum dots and two electrons. There are various kinds of QCA cells proposed which include a four-dot QCA cell and an eight-dot QCA cell. In a QCA Cell-l, two electrons occupy diagonally opposite dots in the cell due to mutual repulsion of like charges. An example of a simple un polarized QCA cell consisting of four quantum dots arranged in a square is as shown in Fig.1. Dots are simply places where a charge can be localized. There are two extra electrons in the cell those are free to move between the four dots. Tunneling in or out of a cell is suppressed.

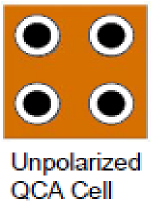


Fig -1: Simple 4-dot Un polarized QCA cell.

NOVEL QCA ADDER

To introduce the novel architecture proposed for implementing ripple adders in QCA, let consider two *n*-bit addends *A* = *an*−1, . . . , *a*0 and *B* = *bn*−1, . . . , *b*0 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)th bit position, the carry signal *ci*+2, furnished at the (*i*+1)th bit position. In this way, the RCA action, needed to propagate the carry *ci* through the two subsequent bit positions, requires only one MG. Conversely, conventional circuits operating in the RCA fashion, namely the RCA and the CFA, require two cascaded MGs to perform the same operation. In other words, an RCA adder designed as proposed has a worst case path almost halved with respect to the conventional RCA and CFA. CLA improve the speed by reducing the amount of time required to determined carry bits. It can be constructed with the simpler, but usually slower. Ripple carry adder (RCA) since each carry bit ripples into the next adder. it's relatively slow since it has to wait for the carry bit to be calculated from the previous adder before which it can't proceed any further in the computation.

Novel 2-bit module shown in Fig. 2 that also shows the computation of the carry *ci*+1 = *M(pi gi ci)*. The proposed *n*-bit adder is then implemented by cascading *n/*2 2-bit modules as shown in Fig. 2(a). Having assumed that the carry-in of the adder is *cin*= 0, the signal *p*0 is not required and the 2-bit module used at the least significant bit position is simplified. The sum bits are finally computed as shown in Fig. 2(b).

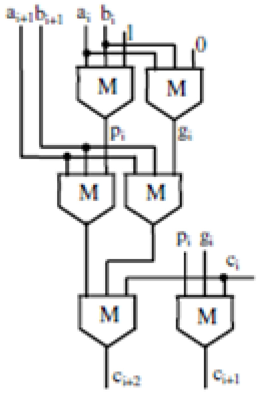


Fig- 2(a): carry of 2- bit qca

It must be noted that the time critical addition is performed when a carry is generated at the least significant bit position (i.e., g0 = 1) and then it is propagated through the subsequent bit positions to the most significant one. In this case, the first 2-bit module computes *c*2, contributing to the worst case computational path with two cascaded MGs.

The subsequent 2-bit modules contribute with only one MG each, thus introducing a total number of cascaded MGs equal to*(n* − 2*)*/2.

Considering that further two MGs and one inverter are required to compute the sum bits, the worst case path of the novel adder consists of *(n/*2) + 3 MGs and one inverter

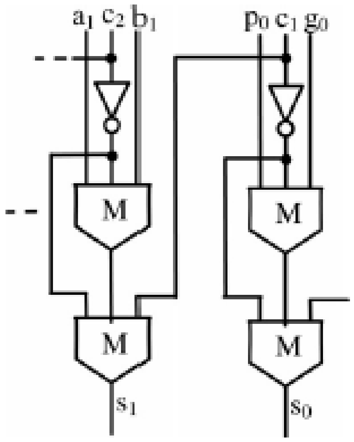


Fig- 2(b) : sum bits of 2-bit qca

The simulation result for the 128-bit adder is shown in Fig. 4. There, the carry out bit is included in the output sum bus. Because of the limited QCA Designer graphical capability, input and output busses are split into two separate more significant and less significant busses. Fig. 4 shows the polarization values of few single output signals (i.e., *sum*128, *sum*64). Simulations performed on 128-bit and 64-bit adders have shown that the first valid result is outputted after five and nine latency clock cycles, respectively. The number of cascaded MGs within the worst case computational path directly impacts on the achieved speed performances as an MG always adds one more clock phase.

However, it is worth noting that because of their different basic logics, designs with the same critical path can achieve different numbers of clock phases. As an example, the novel adder requires less clock phases than the CFA .

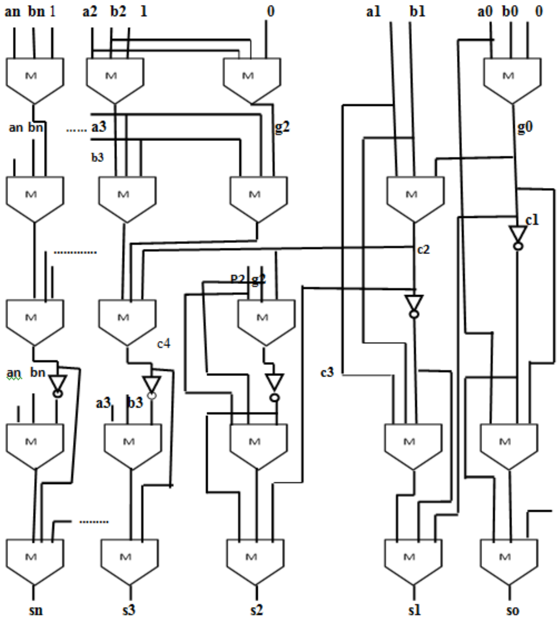
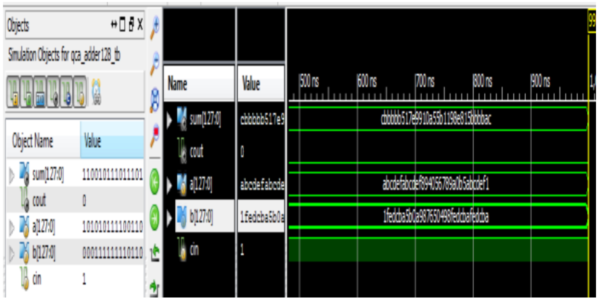


Fig-3:n-bit adder


Objects
Simulation Objects for qca_adder128_tb

Object Name	Value
sum[127:0]	11001011011101
cout	0
a[127:0]	10101011100110
b[127:0]	00011111110110
cin	1

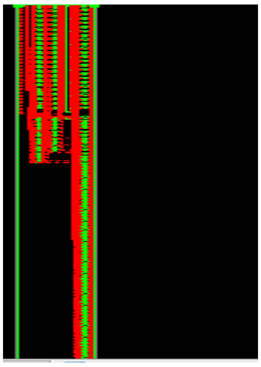
Name	Value
sum[127:0]	cbbbbb517e9
cout	0
a[127:0]	abcdefabcde
b[127:0]	1fedcba5b0a
cin	1

500 ns
600 ns
700 ns
800 ns
900 ns
cbbbbb517e9910a55b1198e815bbbbac
abcdefabcdef894056789a0b5abcdef1
1fedcba5b0a987650498fedcbafedcba

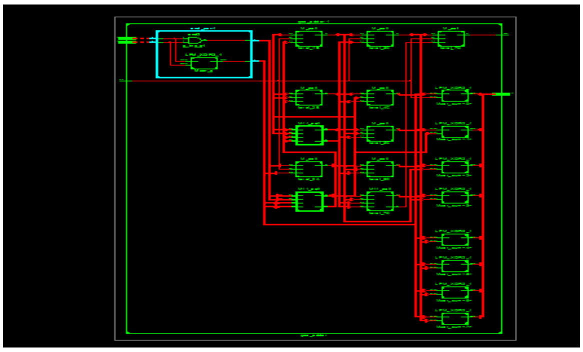
Fig-4:128-bit adder

It should also be noted that the critical path of the HYBA [9] contains the fewest MGs, while the novel adder, the RCA [8] and the CFA [12] require less additional clock phases exceeding the number of cascaded MGs. Results for operands word lengths ranging from 8- to 128-bit also show that the novel adder achieves the lowest delay and spans over an area similar to that occupied by the cheaper designs known in literature. Therefore, our design approach allows the best area-delay tradeoff to be achieved

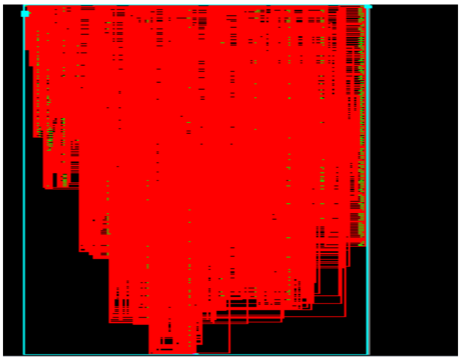
Results: RTL Schematic result of 128-bit



RTL Schematic result of 4-bit



Technical Schematic result of 128-bit



Technical Schematic result of 4-bit

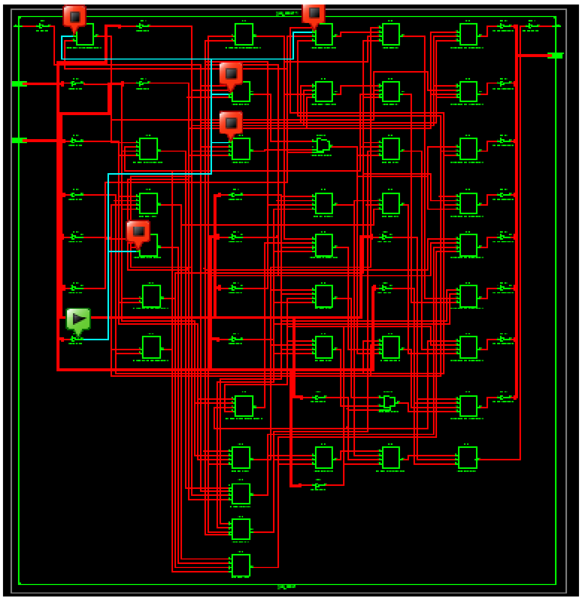


Table -1: Comparison table of 128-bit QCA and RCA

	QCA	RCA
Area	129	256
Delay	18.775ns	140.517ns

CONCLUSIONS AND FUTURE SCOPE

A new adder designed in QCA was presented. It accomplished pace exhibitions higher than all the current QCA adders, with a range prerequisite practically identical with the reduced RCA and CFA. Furthermore, due to the embraced fundamental rationale and format system, the quantity of clock cycles required for finishing the elaboration was restricted. Thus by using the implemented adder we have reduced area and delay when compared to existing ripple carry adder (RCA). A 128-bit adder designed as descried a area of 18.77 ns and gate count of 1336.

References

[1] C. S. Lent, P. D. Tougaw, W. Porod, and G. H. Bernestein, “Quantum cellular automata,” *Nanotechnology*, vol. 4, no. 1, pp. 49–57, 1993.

[2] M. T. Niemer and P. M. Kogge, “Problems in designing with QCAs: Layout = Timing,” *Int. J. Circuit Theory Appl.*, vol. 29, no. 1, pp. 49–62, 2001.

[3] J. Huang and F. Lombardi, *Design and Test of Digital Circuits by Quantum-Dot Cellular Automata*. Norwood, MA, USA: Artech House, 2007.

[4] W. Liu, L. Lu, M. O’Neill, and E. E. Swartzlander, Jr., “Design rules for quantum-dot cellular automata,” in

Proc. IEEE Int. Symp. Circuits Syst., May 2011, pp. 2361–2364.

[5] K. Kim, K. Wu, and R. Karri, "Toward designing robust QCA architectures in the presence of sneak noise paths," in *Proc. IEEE Design, Autom. Test Eur. Conf. Exhibit.*, Mar. 2005, pp. 1214–1219.

[6] K. Kong, Y. Shang, and R. Lu, "An optimized majority logic synthesis methology for quantum-dot cellular automata," *IEEE Trans. Nanotechnol.*, vol. 9, no. 2, pp. 170–183, Mar. 2010.

[7] K. Navi, M. H. Moaiyeri, R. F. Mirzaee, O. Hashemipour, and B. M. Nezhad, "Two new low-power full adders based on majority-not gates," *Microelectron. J.*, vol. 40, pp. 126–130, Jan. 2009.

[8] H. Cho and E. E. Swartzlander, "Adder design and analyses for quantum-dot cellular automata," *IEEE Trans. Nanotechnol.*, vol. 6, no. 3, pp. 374–383, May 2007.

[9] V. Pudi and K. Sridharan, "Low complexity design of ripple carry and Brent–Kung adders in QCA," *IEEE Trans. Nanotechnol.*, vol. 11, no. 1, pp. 105–119, Jan. 2012.

[10] V. Pudi and K. Sridharan, "Efficient design of a hybrid adder in quantumdot cellular automata," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 19, no. 9, pp. 1535–1548, Sep. 2011.

[11] S. Perri and P. Corsonello, "New methodology for the design of efficien
binary addition in QCA," *IEEE Trans. Nanotechnol.*, vol. 11, no. 6, pp. 1192–1200, Nov. 2012.

[12] V. Pudi and K. Sridharan, "New decomposition theorems on majority logic for low-delay adder designs in quantum dot cellular automata," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 59, no. 10, pp. 678–682, Oct. 2012.