IJECEC INTERNATIONAL JOURNAL OF ELECTRICAL AND COMPUTER COMMUNICATION ENGINEERING
ISSN(Online):2533-8945
VOLUME 5 ISSUE 6

DESIGN OF APPROXIMATE ADDERS USING QUANTOM DOT CELLULAR AUTOMATA

T.DILEEP[1], NAVEEN BAVANA[2]

[1]M. Tech Student, VLSI SYSTEM DESIGN, Sri Sivani College of Engineering

[2]Assistant Professor, Dept. of ECE, Sri Sivani College of Engineering

Abstract—As a new paradigm in the nanoscale technologies, approximate computing enables error tolerance in the computational process; it has also emerged as a low power design methodology for arithmetic circuits. Majority logic (ML) is applicable to many emerging technologies and its basic building block (the 3-input majority voter) has been extensively used in digital circuit design. In this paper, we propose the design of an onebit approximate full adder based on majority logic. Furthermore, multi-bit approximate full adders are also proposed and studied; the application of these designs to quantum-dot cellular automata (QCA) is also presented as an example. The designs are evaluated using hardware metrics (including delay and area) as well as error metrics. Compared with other circuits found in the technical literature, the optimal designs are found to offer superior performance.

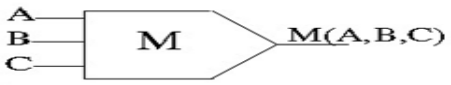
1. INTRODUCTION

Quantum-dot cellular automata (QCA) are an attractive emerging technology suitable for the development of ultra dense low-power high-performance digital circuits. Quantum-dot cellular automata (QCA) which employs array of coupled quantum dots to implement Boolean logic function. The advantage of QCA lies in the extremely high packing densities possible due to the small size of the dots, the simplified interconnection, and the extremely low power delay product. A basic QCA cell consists of four quantum dots in a square array coupled by tunnel barriers. Electrons are able to tunnel between the dots, but cannot leave the cell. If two excess electrons are placed in the cell, Coulomb repulsion will force the electrons to dots on opposite corners. There are thus two energetically equivalent ground state polarizations can be labeled logic "0"and "1".The basic building blocks of the QCA architecture are AND,OR and NOT. By using the Majority gate we can reduce the amount of delay.i.e., by calculating the propagation and generational carries.

2. QCA Majority Gate:

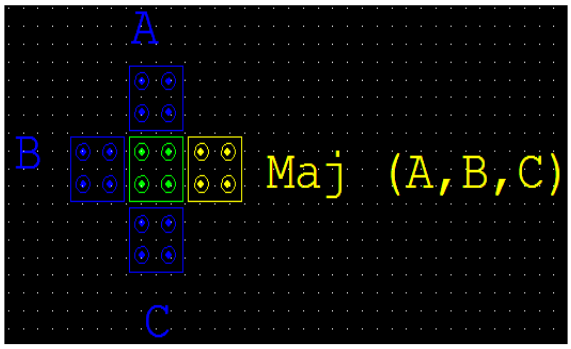
The fundamental QCA logical circuit is the three-input majority logic gate that appears in below figure from which more complex circuits can be built. The basic majority gate is obtained by placing four neighboring cells adjoining to a device cell, which is in the middle. Three of the side cells are used as inputs, while the remaining one is the output. The device cell will always assume the majority polarization because it is this polarization where electron repulsion between the electrons in the three input cells and the device cell will be at a minimum. The logic function implemented by the MV is

Consider the Coulombic cooperation between cells 1 and 4, cells 2 and 4, and cells 3 and 4. Coulombic connection between electrons in cells 1 and 4 would typically bring about cell 4 changing its polarization in light of electron aversion (accepting cell 1 is an info cell). Notwithstanding, cells 2 and 3 additionally impact the polarization of cell 4 and have polarization P=+1. Therefore, on the grounds that most of the cells impacting the gadget cell have polarization P=+1, it too will likewise accept this polarization on the grounds that the powers of Columbic collaboration are more grounded for it than for P=-1. The QCA 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 =AB+BC+CA.



(a)

ISSN(Online):2533-8945
VOLUME 5 ISSUE 6



(b)

Fig 1: Majority gate (a)Logic symbol (c)QCA layout

(e) Simulation output

3 PROPOSED APPROXIMATE FULL ADDERS:

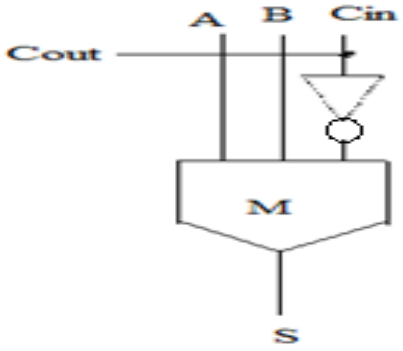
In this section, a one-bit approximate full adder is proposed, which is presented and compared with one-bit accurate full adder and an existing one-bit approximate full adder. In accurate full adder, we have 3 majority gates and 2 inverters while in, existing one-bit approximate full adder AFA1, 2 majority gates and one inverter is used.

Proposed One-bit Approximate Full Adder:

Inspired by AFA1, we propose a new one-bit approximate full adder, namely, AFA2, which saves 2 majority gates, inverter and 0.25 clock cycles of delay and improves the area of the design by up to 72% compared to accurate full adder and moreover, AFA2 has a smaller area than AFA1 in QCA. Consider the truth table in Table I, C is nearly the same as C except in two of the 8 input combinations. Therefore, C can be approximately considered as C to save a majority gate compared with the one-bit accurate full adder when computing the carry out of a one-bit full adder.

$C_{out} = C$

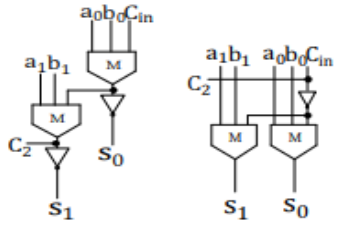
Based on above equation, the inexact output C is substituted to find the approximate output S



Inputs			AFA2	
A	**B**	**C**	**Cout**	**S**
0	0	0	0	0
0	0	1	1	0
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	O	1	1	0
1	1	0	0	1
1	1	1	1	1

Fig. 2 Proposed one-bit approximate full adder: (a) schematic of AFA2 (b) Truth table

PROPOSED TWO-BIT APPROXIMATE FULL ADDERS :



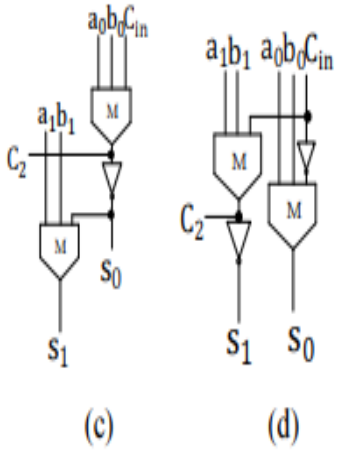


Fig. 3 Schematics of proposed 2-bit approximate full adders: (a) AFA11, (b) AFA22, (c) AFA12 and (d) AFA21.

The inputs to the two-bit adder are given by a = aa, b = bb, c,while s = ss, and care the outputs. By cascading two one-bit approximate full adders (afa1 and afa2), four different combinations are possible for the two-bit approximate full adder; they are shown in fig.2. Afa1 cascaded with afa1 results in the two-bit afa11 design. Similarly, afa2 cascaded with afa2 results in afa22 design. Afa12 consists of afa1 and afa2, in which afa1 is used to compute the lsb; the opposite is applicable to afa21.

The proposed two-bit approximate adders introduce errors for 14 of the 32 input combinations; the MED and NMED of the four approximate adders are provided. The error results show that by cascading two of the same type of one-bit approximate full adder, the MED and NMED are larger than cascading two different types of one-bit approximate full adder; however, afa22 incurs in the smallest area and has less delay than afa12. Considering the number of gates required in an implementation, AFA12 requires one less inverter than AFA21. In terms of the delay, AFA21 needs 0.25 less clocking zones than AFA12.

AFA21 is the best design as it is the closest to the origin. Generally, afa12 and afa21 (with mixed types of one-bit approximate full adders) show better performance compared with those with only a single type of approximate full adders.

Proposed four-bit approximate full adders

Consider a four-bit adder with inputs given by a = $a_0a_1a_2a_3$, b = $b_0b_1b_2b_3$, cand outputs given by s = $s_0s_1s_2s_3$, c. Similar to the two-bit approximate full adder, we can design a four-bit approximate full adder by cascading two two-bit approximate full adders. AFA12 and AFA21 are selected from these two two-bit approximate full adders as these designs show better overall performance than the other two schemes. The proposed designs require fewer gates than an accurate full adder, but at the cost of a reduced accuracy. An improvement of up to 50% in delay and up to 67% in area is achieved. Although AFA1221 has advantages in terms of the reduced number of gates and delay, its MED/NMED is the largest. AFA2121 and AFA2112 have the same MED/NMED, but AFA2121 has less delay. Compared with AFA2112, AFA1212 requires one less inverter with a reduction in med.

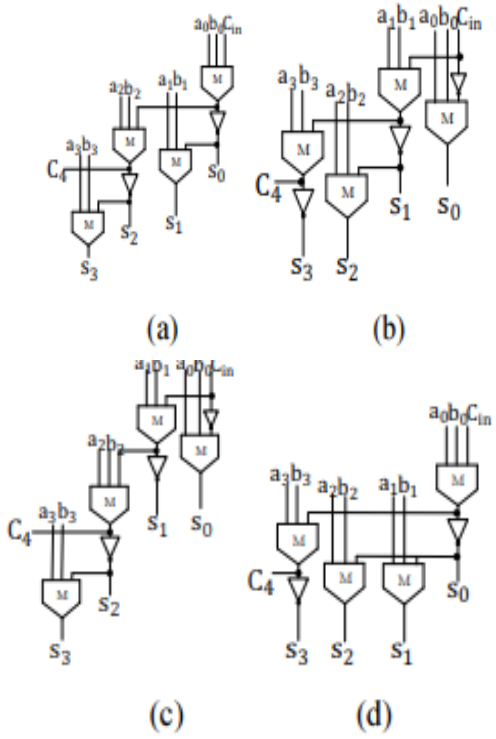


Fig. 4. Schematics of proposed 4-bit approximate full adders: (a) AFA1212, (b) AFA2121, (c) AFA2112 and (d) AFA1221.

For four-bit designs, the schemes in which two of the same type of the proposed two-bit approximate full adders are cascaded have better

performance than cascading different types of approximate full adders.

Proposed eight-bit approximate full adders

Consider an eight-bit adder with inputs, we have designed eight-bit approximate adders by cascading two four-bit approximate adders by using AFA1212 and AFA2121, as they show better overall performance than the other two designs. The proposed eight-bit approximate adders are shown in figure. The proposed designs significantly reduce the number of gates and delay but at the cost of a decrease in accuracy. In terms of gates, AFA1212-1212 and AFA1212-2121 require one less inverter than the other adders; AFA2121-2121 and AFA1212-2121 incur less delay than the other adders.

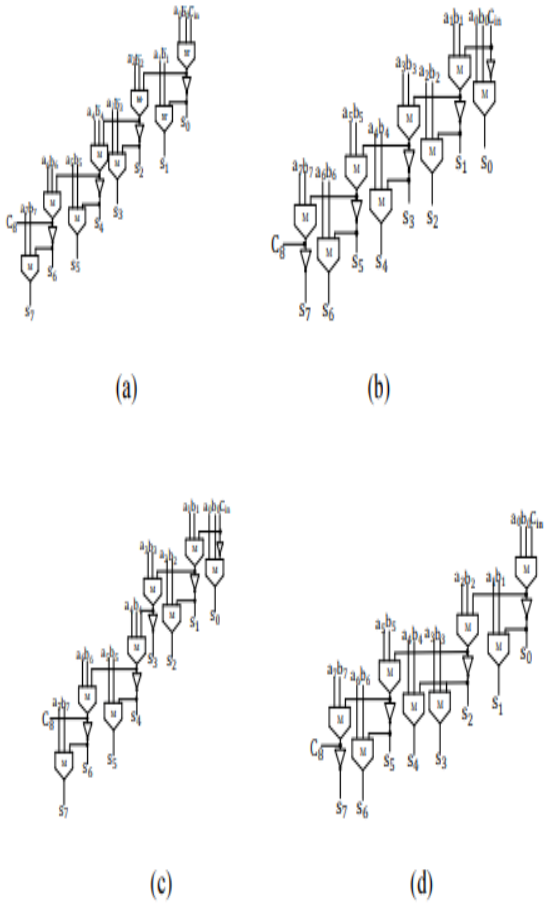


Fig 5. Schematics of proposed 8-bit approximate full adders: (a) AFA1212- 1212, (b) AFA2121-2121, (c) AFA2121-1212 and (d) AFA1212-2121.

4. SIMULATION RESULTS

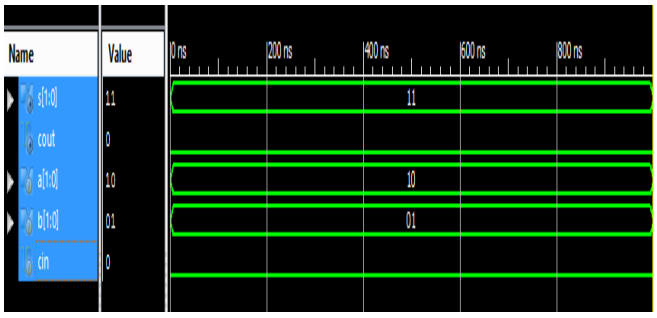


Fig 4.1 AFA 11

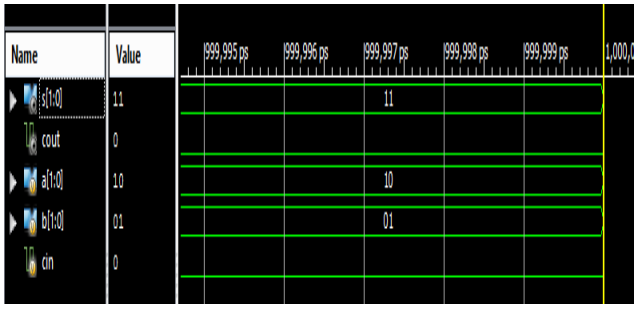


Fig4.2 AFA 22

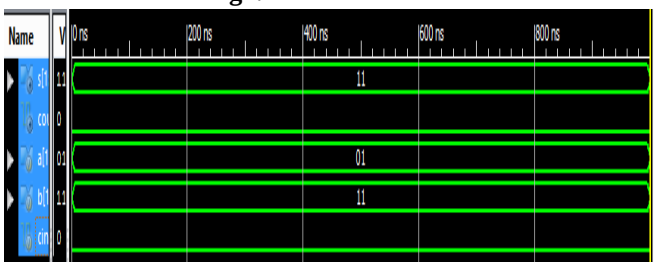


Fig 4.3 AFA 21

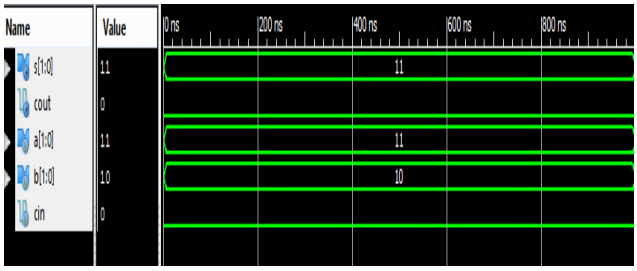


Fig 4.4 AFA 12

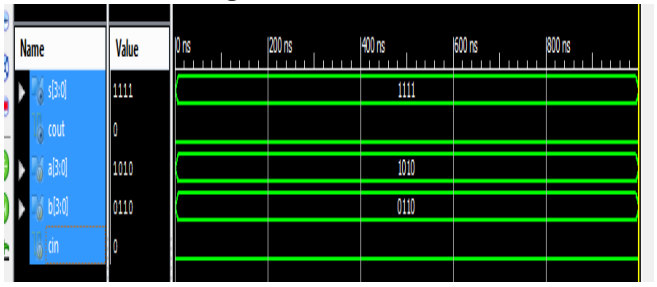


Fig 4.5 AFA 1212

ISSN(Online):2533-8945
VOLUME 5 ISSUE 6

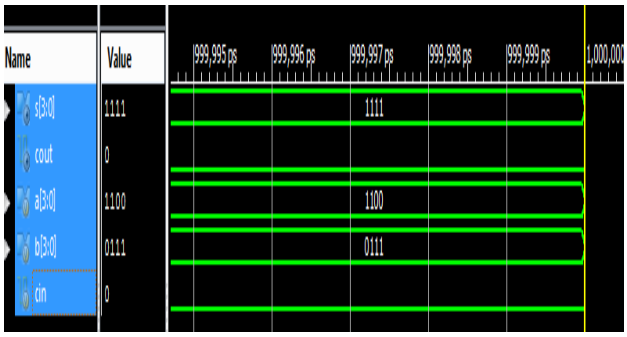


Fig 4.6 AFA 1221

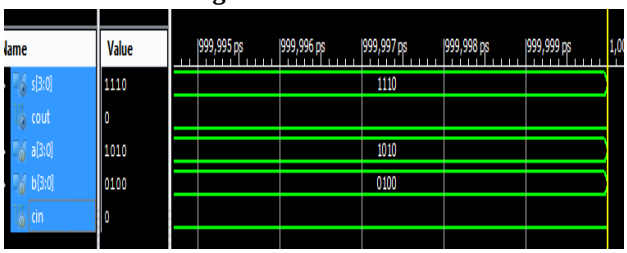


Fig 4.7 AFA 2112

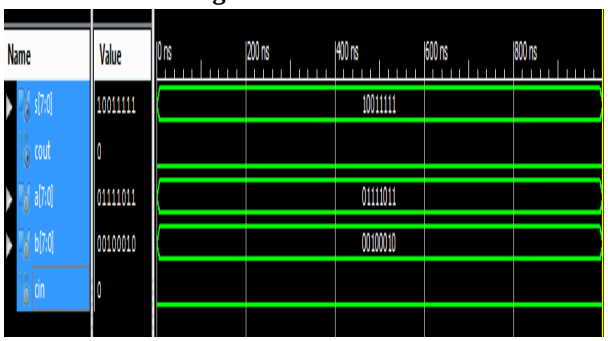


Fig 6.7 AFA12122121

The proposed QCA approximate full adders are designed and simulated by using the QCA Designer tool for the one-bit case. QCA Designer is a QCA layout and simulation tool developed at the University of Calgary. The design and simulation is as follows. First, we generate the layout of the proposed one-bit approximate full adder. Then, we design multi-bit adders using the one-bit layout.

CONCLUSION

Here I have proposed ML based one-bit and multi-bit approximate full adders; these designs show considerable savings in area, delay and number of gates while only incurring a modest loss in accuracy. Compared with the accurate full adder, the proposed designs result in an improvement of at least up to 50% in delay and up to 67% in area for the 4-bit design. An improvement of at least up to 50% in delay and up to 71% in area is achieved for the 8-bit scheme.

The proposed QCA one-bit approximate full adder AFA2 has 13 cells and its outputs are generated after the 0.75 of a clock time period. The proposed full adder is simulated using the QCA Designer 2.0.3 simulation tool and has been compared with AFA1. The simulation results show that the proposed QCA full adder in terms of the number of used cells and occupied area is so better than others. Further, the usefulness of such design is established with the synthesis of high-level logic. Experimental results illustrate the significant improvements in design level in terms of circuit area, cell count, and clock compared to that of conventional design approaches.

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]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.

[3] J. Han and M. Orshansky, "Approximate computing: an emerging paradigm for energy-efficient design," in Proc. ETS, pp. 1-6, May2013

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

[5] S.-L. Lu, "Speeding up processing with approximation circuits," Computer, vol. 37, no. 3, pp. 67–73, Mar. 2004.

[6] C. Labrado, H. Thapliyal and F. Lombardi "Design of Majority Logic Based ApproximateArithmetic Circuits," inProc. IEEE International Symposium on Circuits and Systems (ISCAS), pp. 2122-2125, May 2017.

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

[9] V. Gupta, D. Mohapatra, S. P. Park, A. Raghunathan, and K. Roy, "Impact: Imprecise adders for low-power approximate computing,"in Proc.

ISSN(Online):2533-8945
VOLUME 5 ISSUE 6

Int.Symp. Low Power Electronics and Design (ISLPED) , pp. 409– 414, Aug. 2011.

[10] V. Gupta, D. Mohapatra, A. Raghunathan, and K. Roy. "Low-power digital signal processing using approximate adders,"IEEE Trans. Comput.-Aided Des. Integ. Circuits Syst, vol. 32,pp. 124–137, 2013.