JES
Journal of Engineering Sciences

Vol 10, Issue 12, DEC/2019
ISSN NO:0377-9254

IMPLEMENTATION OF APPROXIMATE FULL ADDERS USING MAJORITY LOGIC

[1]ARIFA PARVEEN PALLEVALE, [2]N. MD. BILAL, [3]S. MOHAN DAS
[1]M.Tech Student, [2]Assistant Professor, [3]Associate Professor
DEPT OF ECE
SVR Engineering College, Nandyal

ABSTRACT

As a new paradigm in the nano-scale 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 a one- bit 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.

I. INTRODUCTION

QCA(Quantum dot Cellular Automata)

Quantum-dot cellular automata (QCA) is 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.

1.1. IDEA OF DESIGNING:

Quantum Dot Cellular Automata (sometimes referred to simply as quantum cellular automata, or QCA) are proposed models of quantum computation, which have been devised in analogy to conventional models of cellular automata introduced by von Neumann. Standard solid state QCA cell design considers the distance between quantum dots to be about 20 nm, and a distance between cells of about 60 nm. Just like any CA, Quantum-dot Cellular Automata are based on the simple interaction rules between cells placed on a grid. A QCA cell is constructed from four quantum dots arranged in a square pattern. These quantum dots are sites electrons can occupy by tunneling to them.

1.2. CELL DESIGN:

The basic device in QCA is a QCA cell which enables both the computation and transmission of the information. A QCA cell consists of a hypothetical square space with four electronic sites and two electrons.

The electronics sites, called Dots, represent the locations which the electrons can occupy. The dots are coupled through quantum mechanical tunneling barriers and electrons can tunnel through them depending on the state of the system. Exactly two mobile electrons are loaded in the cell and can move to different quantum dots in the QCA cell by means of electron tunneling.

Columbic repulsion will cause the electrons to occupy only the corners of the QCA cell resulting in two specific polarizations. Electron tunneling is assumed to be completely controllable by potential barriers (that would exist underneath the cell) that can be raised and lowered between adjacent QCA cells by means of capacitive plates.

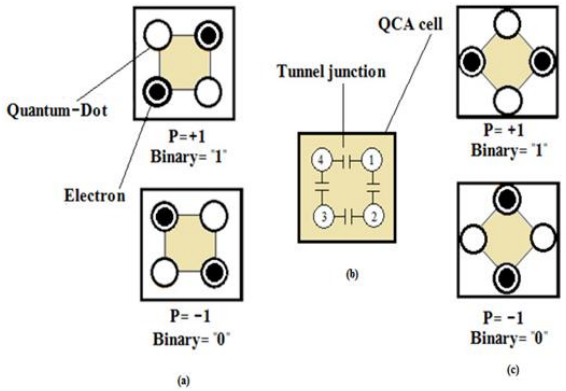


Fig. 1.Two kinds of QCA cells (a) 90° QCA cells (b) QCA cell with tunnel junction (c) 45° QCA cells

For an isolated cell there are two energetically minimal equivalent arrangements of the two electrons in the QCA cell, denoted cell polarization P = +1 and cell polarization P = -1. Cell polarization P = +1 represents a binary '1', while cell polarization P = -1 represents a binary 0. It is also worth noting that there is an unpolarized state as well. In an unpolarized state, inter-dot potential barriers are lowered which reduces the confinement of the electrons on the individual quantum dots.

Consequently, the cells exhibit little or no polarization and the two-electron wave functions have delocalized across the cell. The numbering of dots denoted by i in the cell goes clockwise starting from the dot on the top right with i =1, bottom right dot i =2, bottom left dot i =3, and top left dot i=4. The polarization measures the charge configuration i.e. the extent to which the electronic charge is distributed among the four dots.

1.3. QCA LOGIC DEVICES:

The fundamental QCA logic devices are the QCA wire, majority gate and inverter.

1.4.QCA wire: In a QCA wire, the binary signal propagates from input to output because of the Columbic connections between cells. This is a result of the system attempting to settle to a ground state. Any cells along the wire that are anti-polarized to the input would be at a high energy level, and would soon settle to the correct ground st ate.

A parallel wire is regularly separated into different clock zones, to guarantee that the flag doesn't fall apart as signs for the most part have a tendency to debase with a long chain of cells in a similar timing zone.

Fig .2: A QCA wire composed of coupled cells

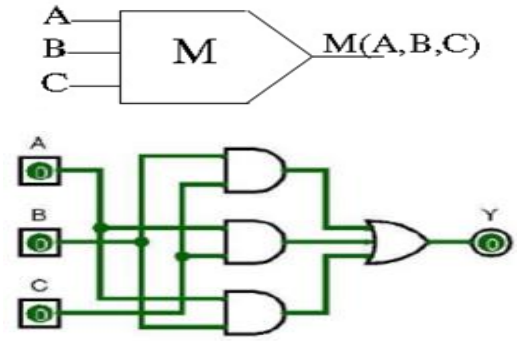
1.5. QCA Majority Gate:

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

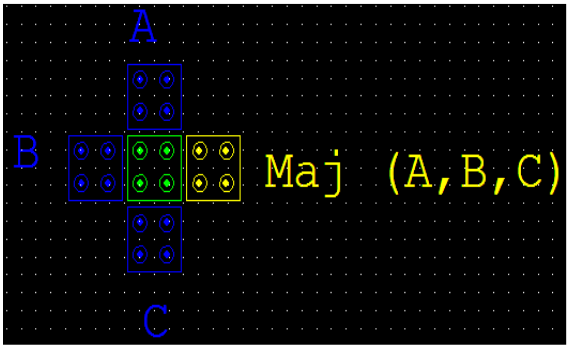
Consider the Coulombic cooperation between cells 1 and 4, cells 2 and 4, and cells 3 and 4. Coulombic connection between electrons in cells 1 and 4 would typically bring about cell 4 changing its polarization in light of electron aversion (accepting cell 1 is an info cell). Not withstanding, cells 2 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.

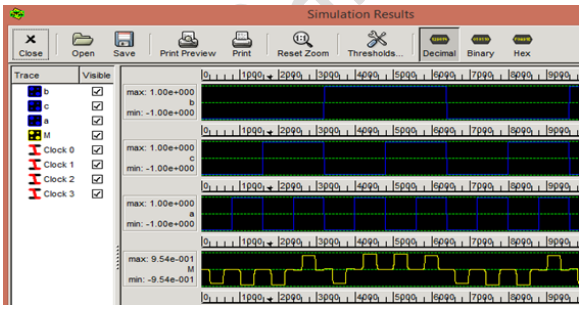


(a)



A	B	C	F
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

(c) **(d)**



(e)

Fig .3: Majority gate (a)Logic symbol (b)Logic circuit (c)QCA layout (d)Truth table

(e) Simulation output

II. LITERATURE SURVEY

An adder is a digital logic circuit in electronics that implements addition of numbers. In many computers and other types of processors, adders are used to calculate addresses, similar operations and table indices in the ALU and also in other parts of the processors.

Various QCA full-adders have been presented to date. The first one (presented in 1994) is composed of five three-input majority gates and three inverters (Tougaw and Lent, 1994). The schematic of this design is shown in Figure

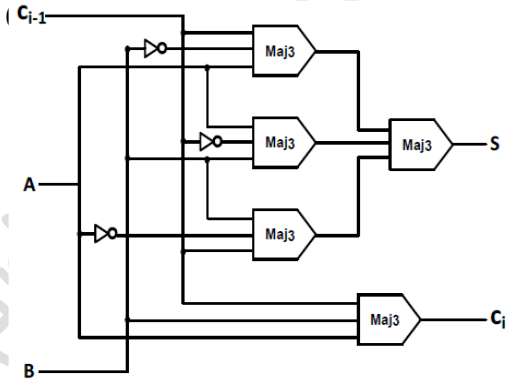


Fig. 4: The QCA full-adder schematic presented in (Tougaw and Lent, 1994).

Hashemi et al. 181 is implemented in one layer using 192 QCA cells. In this design QCA clocking concepts are not considered. In Vetteth et al. (2002) another QCA full-adder using the same logical structure and coplanar wire crossing scheme is presented. In contrast to the previous design, this full-adder incorporates QCA clocking scheme and takes 14 clock phases (3.5 clock cycles) to generate outputs. This full-adder was used in designing a 4-bit CLA (Vetteth et al., 2002).

A simpler QCA full-adder is presented in (Wang et al., 2003). This full-adder is composed of three three-input majority gates and two inverters. It takes 5 clock phases (1.25 clock cycles) to produce outputs. Hence, it is faster than presented design in (Vetteth et al., 2002).

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.

III. PROPOSED METHOD

Majority gate-based logic is not normally explored with standard CMOS technologies, primarily because of the hardware inefficiencies in creating majority gates. As a result, not much effort has been made towards the optimization of circuits based on majority gates. We are exploring one particular emerging technology, quantum-dot cellular automata (QCA), in which the majority gate is the fundamental logic primitive. We report a simple and intuitive method for reduction of three-variable Boolean functions into a simplified majority representation. The method is based on Karnaugh maps (K-maps), used for the simplification of Boolean functions.

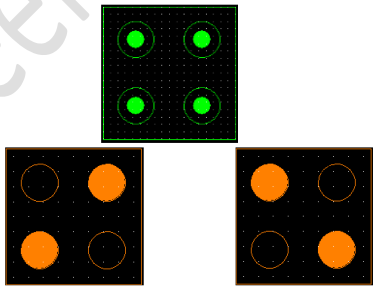
Quantum-dot Cellular Automata (QCA) technology is a solution for implementation of the nanometer sized circuits and it can be a suitable replacement for CMOS. Similar to CMOS technology, designing the basic computational element such as adder with the QCA technology is regarded as one of the most important issues that extensive researches have been done about it.

Decreasing the size of the CMOS transistors is very difficult at nanometer scale. The QCA technology can be viewed as an alternative technology. Dense Nanometer sized circuits can be designed with low power consumption and low delay time in the QCA scope. The QCA circuits and their implementation process aren't involved with the conventional lithography and load depletion challenges.

Designers of the QCA circuits try to reduce the delay time, circuits' occupied area and the number of used cell. Generally a 1-bit full adder has three inputs which are *A*, *B* and C_{in} for producing the *Sum* and the C_{out} for outputs. A number of algebra equations can be used to show how the outputs of the full adder are generated. An algebra equation is composed of some algebra operators and some input variables; also each algebra operator points to a QCA gate.

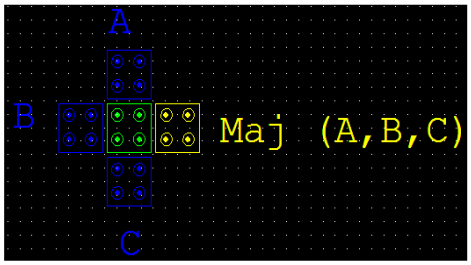
Quantum-dot Cellular Automata :

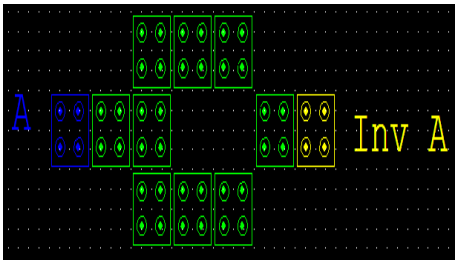
QCA makes use of the polarization state of cells to encode binary information and Columbic force interactions between cells to achieve circuit functionality; these features make this technology substantially different from CMOS. A QCA cell consists in its simplest form of four quantum dots and two electrons that can tunnel between them. Due to Columbic repulsion, electrons are forced to occupy the opposite diagonal vertices (dots). This forms two different polarization states (i.e., -1, +1) for each cell, thus representing logic values of 0 and 1. QCA requires a clocking scheme with four different operational phases, i.e. Switch, Hold, Release, and Relax; each adjacent so-called zone is shifted in phase by 90 degrees to control the flow of information.



(a)

P = +1 **P = - 1**





(c)

Fig.5. QCA basic elements: (a) QCA unpolarized cell, (b)Polarized QCA cells (c) QCA majority gate (voter), and (c) QCA inverter

For a majority gate in QCA, which can be seen in Fig, there is a 0.25 clock delay in each clocking zone. As another basic gate in QCA, an inverter is shown in Fig.

Approximate full adder:

In latest nanotechnology, study on the architecture of nanoscale full adder is considerably enlarged particularly in the QCA.

Error Metrics for Approximate Circuits:

Approximate computing is by definition not correct. For that reason it is important to evaluate approximate computing designs on their accuracy compared to completely correct designs.

As approximate computing introduces errors, error metrics are required to evaluate the accuracy of approximate circuits. In this paper, we evaluate approximate designs using the Mean Error Distance (MED) and the Normalized Mean Error Distance (NMED). The MED is defined as the average of the Error Distance(ED) which is the absolute difference between the approximate and the accurate results across all possible inputs. The NMED is the normalized MED. The definitions of ED, MED and NMED are as follows:

Error Distance(ED) = $|X - Y|$

Mean Error Distance (MED) = $\sum ED / n$

Normalized Mean Error Distance(NMED) = MED / MAX

where X, Y, n and MAX denote the accurate result, the approximate result, the counts of all possible inputs and the maximum value of the result, respectively.

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.

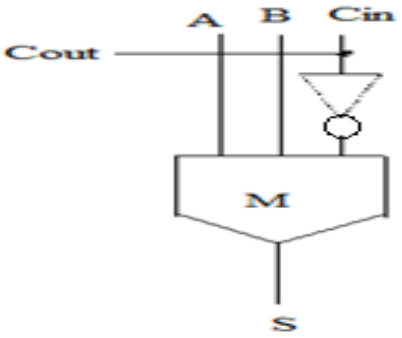
Proposed One-bit Approximate Full Adder:

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

Journal of Engineering Sciences
Vol 10, Issue 12, DEC/2019
ISSN NO:0377-9254

To evaluate the accuracy of the proposed design, the MED and NMED are given by,

$$MED_{AFA2} = \frac{1}{8}(0+1+0+0+0+0+1+0) = 0.25$$

$$NMED_{AFA2} = \frac{MED_{AFA2}}{3} = 0.083$$

Proposed multi-bit approximate adders :

In this section, multi-bit approximate full adders are proposed by merging the proposed and the existing one-bit approximate full adders. Both the corresponding hardware designs and error metrics are evaluated.

Proposed two-bit approximate full adders :

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.6. 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.

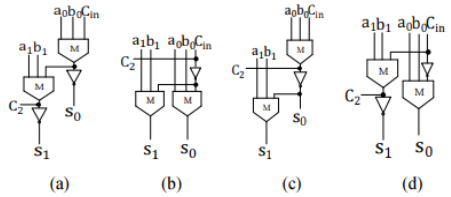


Fig. 7. Schematics of proposed 2-bit approximate full adders: (a) AFA11, (b) AFA22, (c) AFA12 and (d) AFA21.

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 = $b_0b_1b_2b_3$, c and 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.

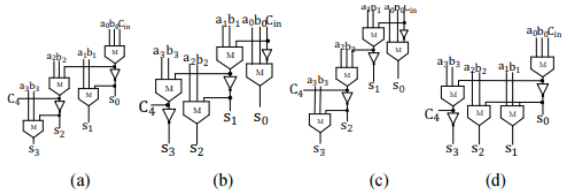


Fig. 8. Schematics of proposed 4-bit approximate full adders: (a) AFA1212, (b) AFA2121, (c) AFA2112 and (d) AFA1221.

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.

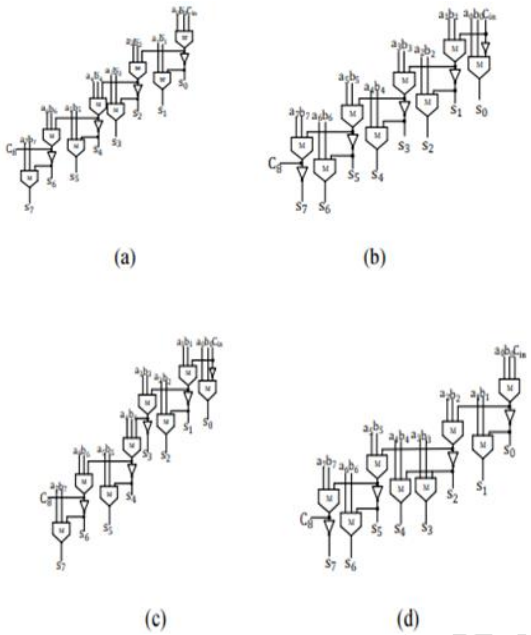
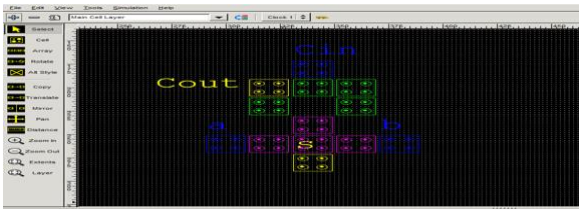


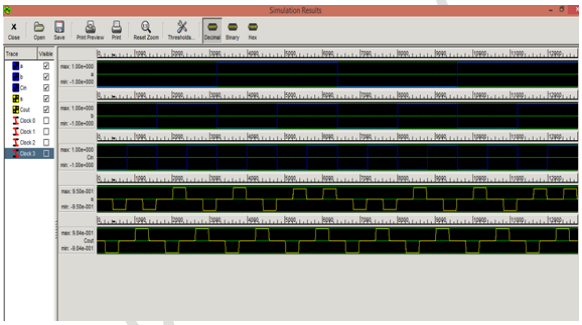
Fig. 9. Schematics of proposed 8-bit approximate full adders: (a) AFA1212- 1212, (b) AFA2121-2121, (c) AFA2121-1212 and (d) AFA1212-2121.

IV. SIMULATION RESULTS

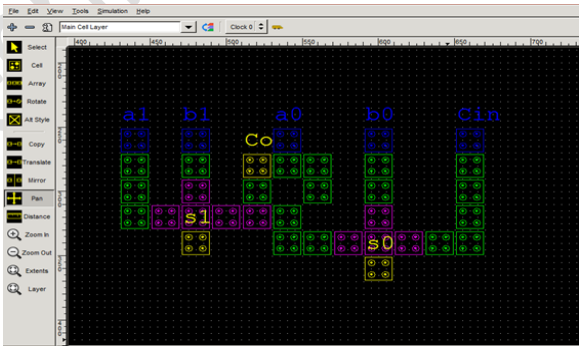
The proposed QCA approximate full adders are designed and simulated by using the QCADesigner tool for the one-bit case. QCADesigner is a QCA layout and simulation tool developed at the University of Calgary. The design and simulation is as follows. First, we generate the layout of the proposed one-bit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



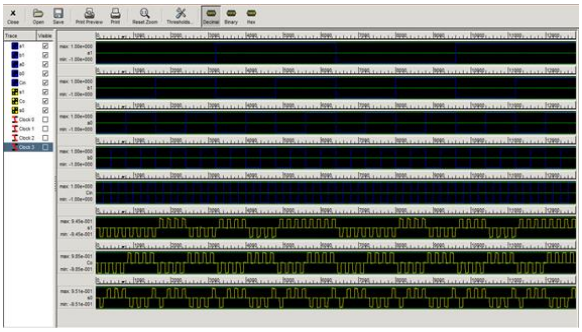
(a) 1- bit approximate full adder layout



(b)1-bit approximate full adder simulation

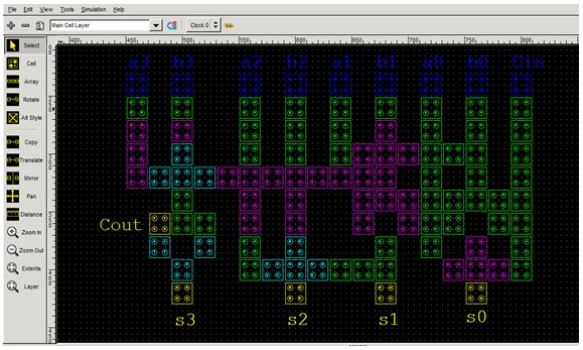


(a) 2-bit approximate full adder layout

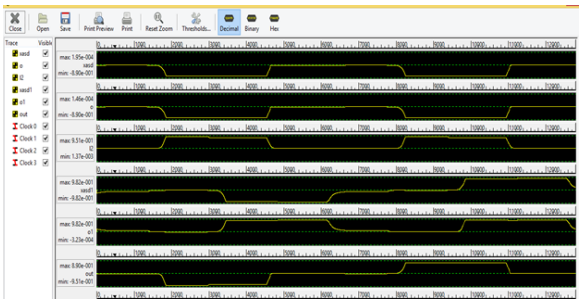


(b) 2-bit approximate adder simulation

Journal of Engineering Sciences
Vol 10, Issue 12, DEC/2019
ISSN NO:0377-9254

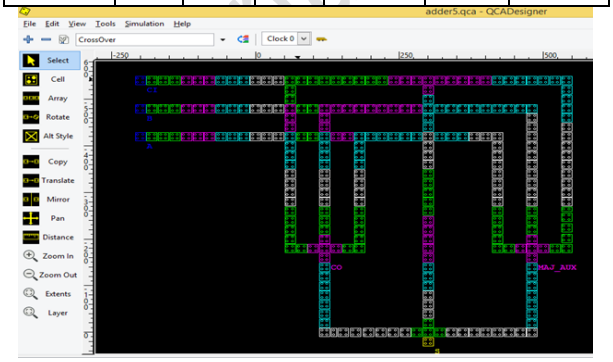


4 bit approximate adder layout

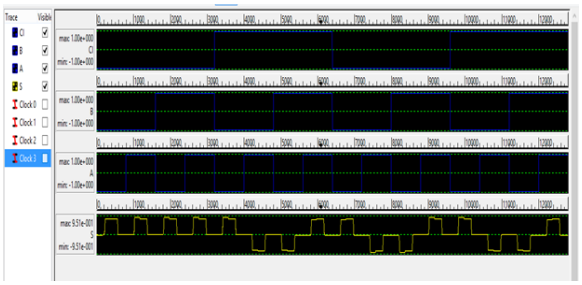


4-bitapproximateaddersimulation

ADDER TYPE	MV	INV	D	A(nm^2)	MED	NMED
AFA11	2	2	0.5	44073	0.75	0.107
AFA22	2	1	0.25	35383	0.75	0.107
AFA12	2	1	0.5	39267	0.625	0.089
AFA21	2	2	0.25	41069	0.625	0.089



8-bitapproximateadderlayout



8-bit approximate full adder simulation

B. Comparison and Discussion

A comparison in terms of number of majority gates (MV), number of inverters (INV), MED, NMED, delay (D) and area (A) between the accurate adder, AFA1 and the proposed approximate full adder AFA2 is reported in Table. Compared with the accurate adder, AFA2 saves 2 majority gates, 1 inverter and 0.25 clock cycles of delay and improves the area of the design by up to 72%. Moreover, AFA2 has a smaller area than AFA1 in QCA.

Table : ONE-BIT APPROXIMATE FULL ADDER COMPARISION

ADDER TYPE	No. of cells	MV	INV	A(nm^2)	MED	NMED
Accurate	14	3	2	60050	0	0
AFA1	13	1	1	18902	0.25	0.083
AFA2	13	1	1	16284	0.25	0.083

Table : TWO-BIT APPROXIMATE FULL ADDER COMPARISION

Table : FOUR-BIT APPROXIMATE FULL ADDER COMPARISION

ADDER TYPE	MV	INV	D	A(nm^2)	MED	NMED
AFA1212	4	2	0.75	76120	2.83	0.910
AFA2121	4	3	0.5	82619	2.87	0.092

AFA2112	4	3	0.75	83936	2.87	0.092
AFA1221	4	2	0.5	82085	5.45	0.175

Table : EIGHT-BIT APPROXIMATE FULL ADDER COMPARISION

ADDER TYPE	**MV**	**INV**	**D**	**A(nm²)**	**MED**	**NMED**
AFA1212-1212	8	4	1.25	184640	46.2	0.090
AFA2121-2121	8	5	1	206425	47.02	0.092
AFA2121-1212	8	5	1.25	218257	46.40	0.091
AFA1212-2121	8	4	1	191126	46.82	0.092

V. 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 Approximate Arithmetic Circuits," in Proc. 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. 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.