Microprocessors and Microsystems 75 (2020) 103040

Contents lists available at ScienceDirect

Microprocessors and Microsystems

journal homepage: www.elsevier.com/locate/micpro

Design of BCD adder with five input majority gate for QCA

V.J. Arulkarthick[a,*], Abinaya Rathinaswamy[a], K. Srihari[b]

[a] Department of ECE, JCT college of Engineering and Technology, Tamilnadu, India
[b] Department of Computer Science and Engineering, SNS College of Engineering, Coimbatore, Tamilnadu, India

ARTICLE INFO

Article history:
Received 25 November 2019
Revised 4 February 2020
Accepted 14 February 2020
Available online 24 February 2020

Keywords:
Microprocessor
Communications
Microsystems

ABSTRACT

In the digital world BCD numbers play a pivotal role in constituting decimal numbers. New different technologies are emerging in order to obtain low area/power/delay factors to replace the CMOS technology. One such technology is quantum cellular automata (QCA) realization, through which many arithmetic circuits can be designed. This paper deals with the implementation of BCD adder with 5 input majority gates for QCA. The 3 input majority gate and an inverter are basic elements of QCA. In this project amalgamation of majority gates with 3 and 5 inputs are used instead of implementing the entire circuit using 3 input majority gate in the BCD i.e. mainly comprised by partly consumed gates and entirely consumed gates. The proposed is designed and functional verification is done by Verilog HDL and Modelsim version 10.4a. The proposed design has been verified and the delay of existing and proposed design is analysed using Xilinx tool. The numbers of partly consumed and entirely consumed gates are less when compared to the existing method of implementation. The delay is reduced compared to the existing system which shows the improvement of 9.84%. The drawback of crossovers that leads the difficulty in implementation and reduces the efficiency of the circuit is reduced in the proposed implementation.

1. Introduction

In natural philosophy adders are used to perform addition of numbers. Adders are used in several computers and different kinds of processors not solely to perform operations in ALUs, however additionally in different elements of the processor for address calculation, signal processing applications and so on [1]. Approximate adders, High speed adders and error tolerant adders are the different adders used for signal processing applications [2].

Even though adders are designed for different numerical formats, the most common adders handles ones and zeros formats. The half adder adds two single bits to produce its corresponding output sum and carry. In the same way the full adder produces two outputs as sum and carries by adding three single bit inputs. Likewise a BCD adder produces a BCD sum by adding the BCD inputs [3].

Assume that the two inputs are labelled A and B with output as sum S and carry C. If the value of A is 0010 in binary equivalent that is 2 in decimal format and B is 0011 in binary equivalent that is 3 in decimal format, then the output sum S and carry C is 0101 and 0 respectively in both binary sum and BCD sum [7]. But when the value of sum is more than 9, 1 will set to the value of carry set and again the sum value will start from 0000 to 1001 with carry 1 for 10 to 19 respectively. The obtained value of sum (if it exceeds 9) should be summed up to the value 6, for the purpose of obtaining the BCD code with 1 as carry.

As the VLSI were developed, monstrous operations and various concepts were proposed in these streams. Even though VLSI technologies also have the advantages such as reduced delay, area and power factors there are some physical limitations in the technology of CMOS [8]. In future the concept of "BEYOND CMOS" can begin because the scaling of the present CMOS technology can reach the basic limit. New technologies are needed to solve the drawback. Emerging device technology will overcome the scaling drawback within the current CMOS technology.

Some of the "Beyond CMOS" technologies are RTD, SET and QCA. A QCA represents quantum-dot cellular automata which is one of the contestant for the fore coming beyond CMOS generation [9].

Quantum-dot cellular automata (QCA) is a paradigm without transistor computation that replaces some complex physical problems caused by the transistors. Quantum dot cellular automata is having cellular arrangements as an array. Quantum dot is a nano

* Corresponding author.
E-mail addresses: Karthick.arul@gmail.com (V.J. Arulkarthick), abinayarathinaswamy@gmail.com (A. Rathinaswamy), hariontо@gmail.com (K. Srihari).

https://doi.org/10.1016/j.micpro.2020.103040
0141-9331/

dimensional semiconductor material with distinguishing property of conductivity depends on the size of the material. The exact nano dimension will be nano metre in size. A fundamental cell of QCA has 4 place holders. Place holders are also called as quantum dots. The quantum dot can either be empty (dot) or filled with electrons. The filled dot (electrons) can only be accommodated in opposite direction in the cell.

The existing systems of the design of decimal full adder entirely consist of majority gate with 3 inputs and NOT gate (inverter). This project deals with replacing the MGs by amalgamating the MGs with 5 and 3 inputs depending on operation requirement which leads to the reduction of crossover complexities, delay and the number of gates.

2. Review of literature

The DFA in QCA, studies various proposals.

Keivan Naviet al in [5], presented a novel seven input MGs that was verified by connections physically and QCA Designer. A seven input MGs includes seven inputs with one output. This seven input MG is used to obtain basic gates such as AND, OR with four inputs. Accordingly almost all up-to-date planned styles in QCA is simplified through this majority device so as to scale back quality, cell count and latency.

Razieh Farazkish et al. in [6], proposed a novel 5-input majority gate. This new 5 input MG can be achieved by a single layer. It requires solely ten cells of QCA. Similar to the above literature the MGs was verified by the connections physically (physical relations) and simulations. A new design of full adder in QCA technology using MG based on MG with 5 inputs is used for the verification of correctness. This design needs solely 2 majority gates and 2 inverters that are enforced by means that of diagonal cells.

The new Full-Adder and former Full-Adder styles are compared in terms of complexness, space and latency. These designs have been verified in QCA Designer, a QCA simulation and layout verification tool. All the 32 states of inputs are checked in order to verify the accuracy. As the verification was done by the simulation, it shows that the newly designed Full-Adder using the MGs with 5 inputs reflects a simple arrangement. Moreover it requires less number of cells and area compared to the previous designs. The new Full Adder has a supreme configuration compared to other structures so that is the main reason that this design will lead to survive in advanced operations.

Subhashee Basu in [13] proposed the realization of gates XOR as well as its complementary gate (XNOR)through innate properties of fundamental gates. The implemented technique reveals the fact that the absence of expanding the number of cells of the circuitry and size can be achieved when implementing the other fundamental gates efficiently. As it is known that the XNOR gate is the complement of XOR, an inverter can be used to collect the output of XOR as input and produce the complement of the XOR gate as output. Yet the above mentioned for complement operation leads to enhance the area and entanglement.

Gurmohan Singh, R.K. Sarin et al. in [4] expressed his idea that the continuous scaling down of feature size has pushed CMOS technology to approach its practical and theoretical limits. Lot of research efforts at nano scale are in progress to explore alternate viable technologies in coming era of advanced integrated circuits [10]. QCA is emerging as a potential technology that could be used in fore-coming computing circuits/systems replacing existing Silicon technology.

3. Existing system

BCD is the abbreviation of Binary Coded Decimal. A BCD adder is the combinative circuitry that sums 2 Binary Coded Decimal

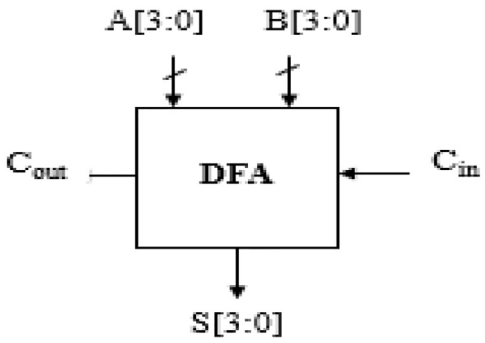


Fig. 1. Abstract view of DFA.

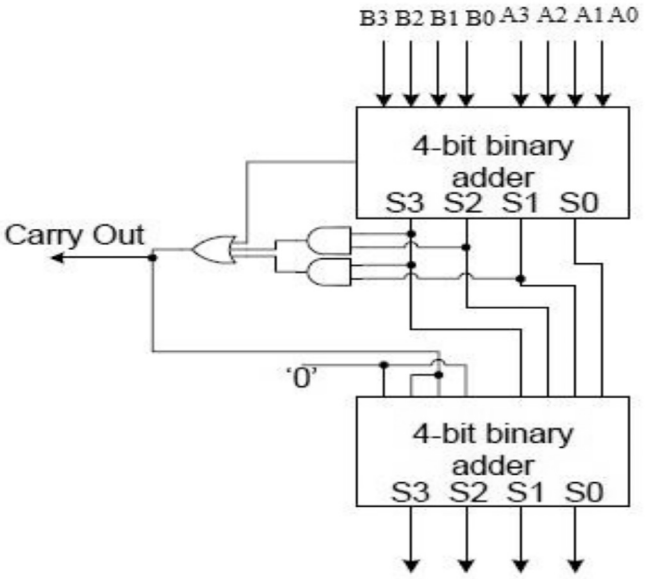


Fig. 2. Block Diagram of Conventional BCD Full Adder.

numbers. BCD may be a category of encryption within which every digit is portrayed by some fixed range of bits. BCD binary numbers represent decimal digits zero to nine. As an illustration [11], (8429)10 can be depicted as (1000 0100 0010 1001) in the view of the fact that every numeral in decimal range will be encoded to 0 s and 1 s and chained together for making this type of illustration. The abstract view of DFA is as follows,

The addition of BCD will be analogous to traditional 0 s and 1saddition. Some excluding situations are, once the total for 2 BCD numerals surpasses nine or a Carry is generated [12], the value six should be summed together with the sum to convert the error range into a legitimate number. Thus a new carry will be created by summing the number six to the inappropriate BCD number that will be moved ahead in such a way that it will be forwarded towards the succeeding digit of the BCD [14].

A BCD 1-digit adder circuit will sums up 2 BCD numbers (digits) concurrently and also generates the output digit in BCD along with required correction logic. A conventional implementation of BCD full adder is given as follows in Fig. 2.

With the intention of examining the output adder of BCD employs a circuitry to check the output of adder placed first. It verifies whether the output surpassed the value nine. In case of some illegitimate conditions the adder placed next will start its function of summing up a six. Appropriate yield will be obtained from the adder other than the first one. In case of legitimate conditions

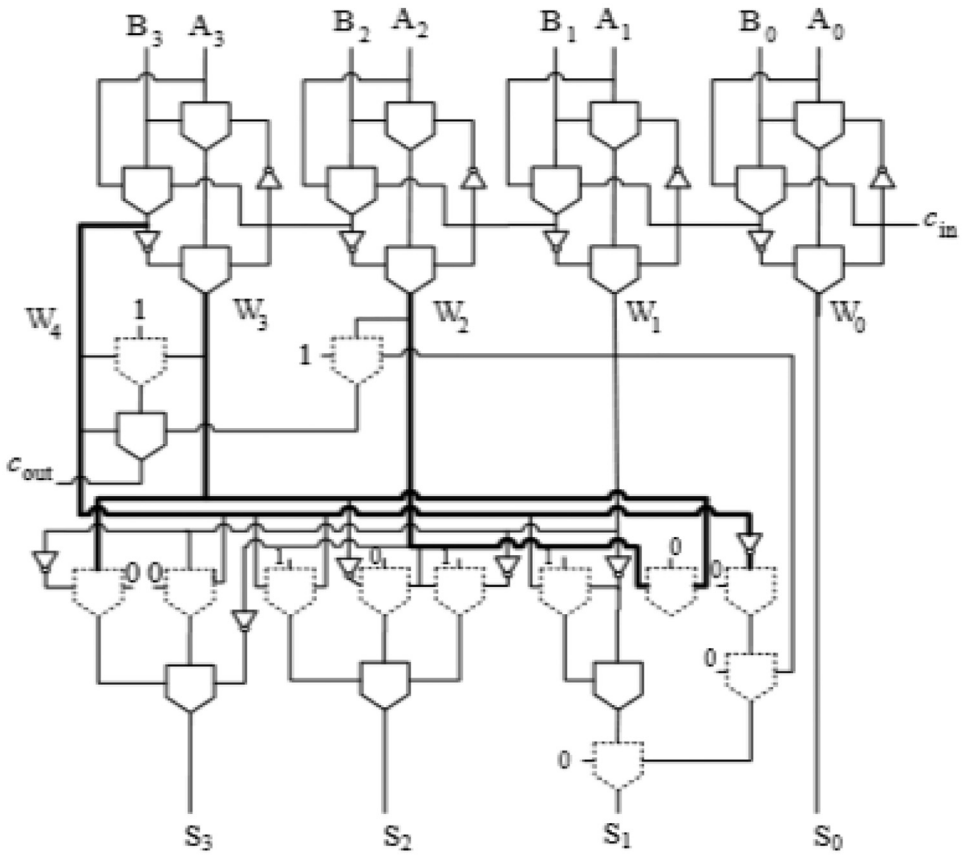


Fig. 3. Implementation of Eqn (8) to (12).

(within nine) 0 can be added so that the yield offered by both the adders are similar.

In existing system, a BCD adder was implemented with completely used MGs as well as PUMs. The existing implementation was verified by QCA Designer. The conventional equations are given as follows,

$$ci = M(gi, pi, ci - 1) \tag{1}$$

$$w3' = g3c1 \tag{2}$$

$$w2' = g3\overline{c1} \vee p3p2(p1 \vee c1) \vee g2g1c1 \tag{3}$$

$$w1' = (M(\overline{c2}, M(g1, p1, \overline{c2}), c1))' \tag{4}$$

$$w0' = M(\overline{c1}, M(g0, p0, \overline{c1}), cin) \tag{5}$$

$$w3 = p3 \vee c3 \tag{6}$$

$$cout = M(g3, p3, p2) \vee (p3 \vee g2)p1 \vee (p3 \vee M(g2, p2g2))c1 \tag{7}$$

$$cout = M(w4, w3 \vee w4, M(w1, w2, w3)) \tag{8}$$

$$s3 = M(\overline{w2}, w4w1, w3\overline{w1}) \tag{9}$$

$$s2 = M(w4 \vee w1, \overline{w3}w2, w2 \vee \overline{w1}) \tag{10}$$

$$s1 = M(\overline{w1}, w4\overline{w1}, w3w2) \vee \overline{w4 \vee w3w1} \tag{11}$$

$$s0 = w0 \tag{12}$$

And the modified equations from the conventional one and the implementation diagram is given in Fig. 3.

To improve the utilization of majority gate and to reduce the problem of crossover complexities, the S1 and S2 equations are further improved as

$$s1 = M(w1 \vee cout, \overline{coutw1}, 0) \tag{13}$$

$$s2 = M(\overline{M(w1, w2, cout)}, cout \vee w2, w1w2) \tag{14}$$

The implementation of the above Eqs. (13) and (14) with modified S1 and S2 is given in Fig. 4.

4. Proposed system

Using the modified equations in existing method and the concept MGs with 5 inputs the proposed method was implemented. The analysis of existing system which uses the MGs with 3 inputs and the proposed system which uses both the MGs with 5-input as well as 3-input has been done through the model sim and Xilinx software. Compared to the existing system the union of MGs with 5 and 3-inputs will reduce the number of partly consumed gates (PUMs). In existing system the number of consumed gates are 9 and the entirely consumed gates are 16. In proposed method the PUMs are 8 and entirely consumed gates are 11. So the total number of gates used in the proposed method also reduces from 25 majority gates to 19. The delay is reduced from 15.582 ns to 14.049 ns. The proposed implementation is given below,

4.1. MGs with five inputs

Primarily various explorers had consumed MGs with 3 inputs towards implementing the circuits of quantum cellular automata. Subsequently the necessity for efficiency increases to a greater extent. As a result MGs with 5 inputs were created. It can be arithmetically given as

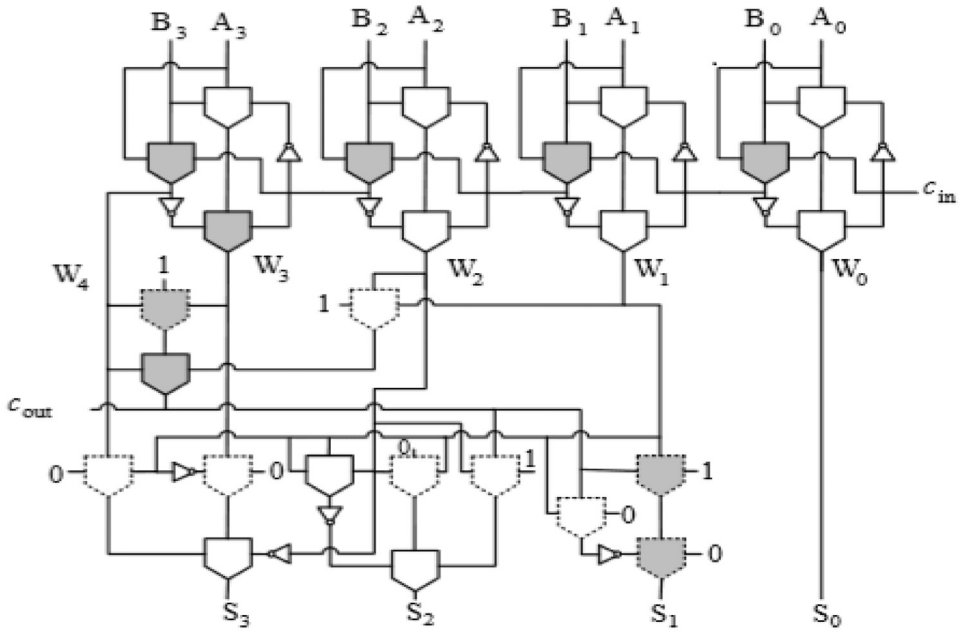


Fig. 4. Implementation of the existing system.

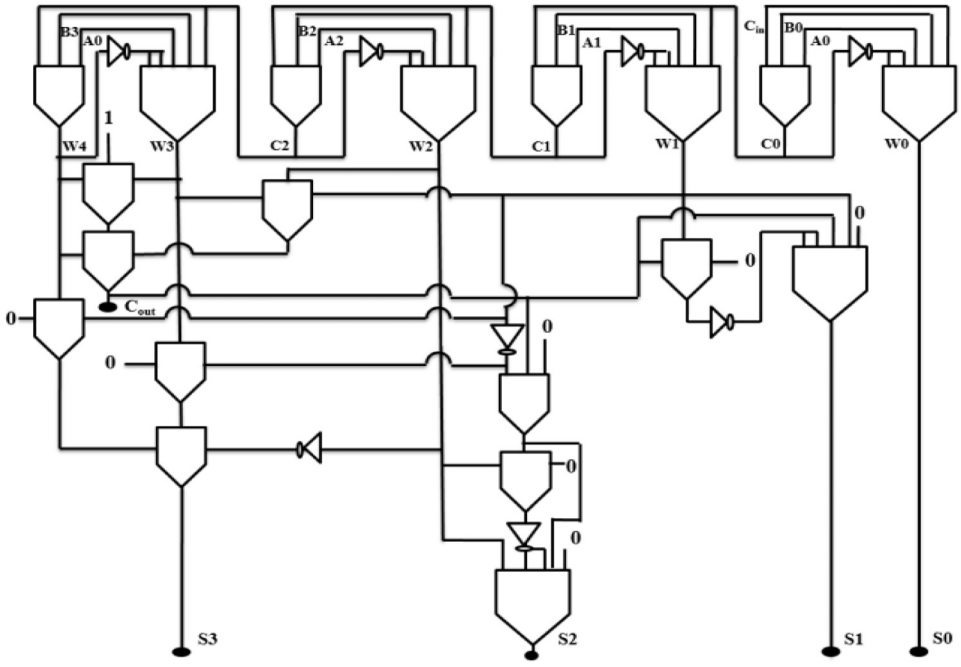


Fig. 5. Implementation of Proposed method.

$$M(A, B, C, D, E) = ABC + ABD + ABE + ACD + ACE + ADE + BCD + BCE + BDE + CDE \quad (15)$$

The equation given below represents the 1 binary digit full adder. It consists of three inputs (X, Y and Cin) and 2 outputs (Z or S, C_0). Each of the yields and ingoing data are one bit.

$$c0 = xy + ycin + cinx \quad (16)$$

$$s = xycin + x'y'cin + x'ycin' + xy'cin' \quad (17)$$

It is necessary to depict the Eqns (16) and (17) with regard to MGs for the implementation of adder(full adder) in QCA technology that can be given as

$$c0 = M3(x, y, cin) \quad (18)$$

$$z = M5(x, y, cin, c0', c0') \quad (19)$$

M3 in the above expression depicts the MG with 3 inputs and M5 depicts MG with 5 inputs. With the help of negator binary addition can be performed which is given below in the generalized equation. MAJ3 and MAJ5 are same as M3 and M5given in the following eqn (20) and (21).

$$si = MAJ5(ai, bi, ci - 1, \overline{ci}, \overline{ci}) \quad (20)$$

$$ci = MAJ3(ai, bi, ci - 1) \quad (21)$$

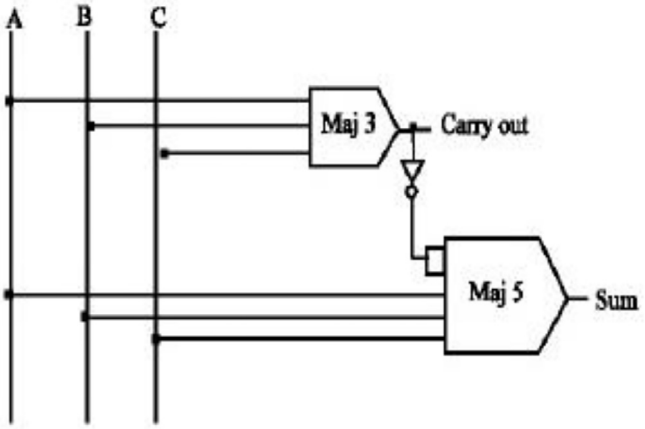
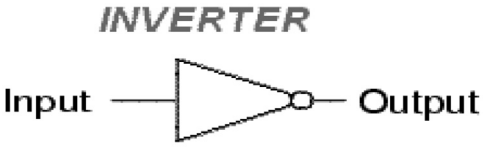


Fig. 6. Diagrammatic representation of adder.

Full adders are used to implement the BCD adder here. Initially To obtain the binary output full adders are used. Then various equation simplifications are done further to obtain binary coded decimal. The diagrammatic representation depicted in Fig. 6 denotes the design of full adder by majority circuits. The model becomes simpler through the utilization of majority gate with 5 inputs compared to the model utilizing solely MG with 3 inputs and negator (inverter).

4.2. Inverter

The inverter in QCA performs as same as the NOT gate functionality in digitalelectronics. It produce the complement of the input. If the input is one, then the output is going to be zero. If the input is zero, then the output is going to be one. The inverter symbol and the truth table is shown in Fig. 7.



Input	Output
1	0
0	1

Fig. 7. Inverter symbol and its truth table.

4.3. XOR using MGs

The circuits especially digital can be implemented with the union of fundamental elements such as logical conjunction (AND), logical disjunction (OR), inverter (NOT) furthermore universal gates negated OR and negated AND. Additionally utilization of exclusive conjunction and exclusive disjunction will provide more benefits to advanced designs. In order to implement the XOR function in QCA the XOR output should be obtained using the MGs.

Ex-Or gate is used to reduce the number of majority gates and implement the equations to achieve the output bit S1 from the outputs of full adders that is w1,w2,w3 and w4. It is very clear that the 6- Majority gate from the Fig. 3 are reduced to single 5 bit majority gate and 1 3-bit majority gate. The carry is used for performing the calculation of MSB.

Instead of transistors, majority gates are used here. This BCD adder is especially for QCA. The QCA supports only the majority gate and inverters. The input is first added using full adder and produces a sum and carry bit. With proper combinations of those sum and carry binary coded decimal formats are obtained as output. Here the significant bit is required to consider the accuracy of

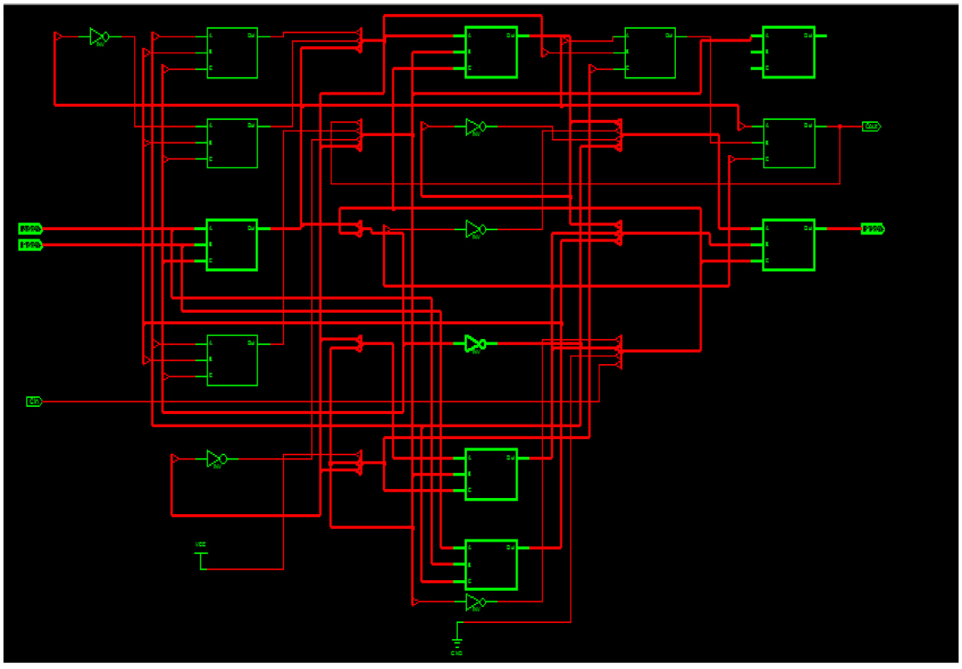


Fig. 8. RTL schematic of existing method.

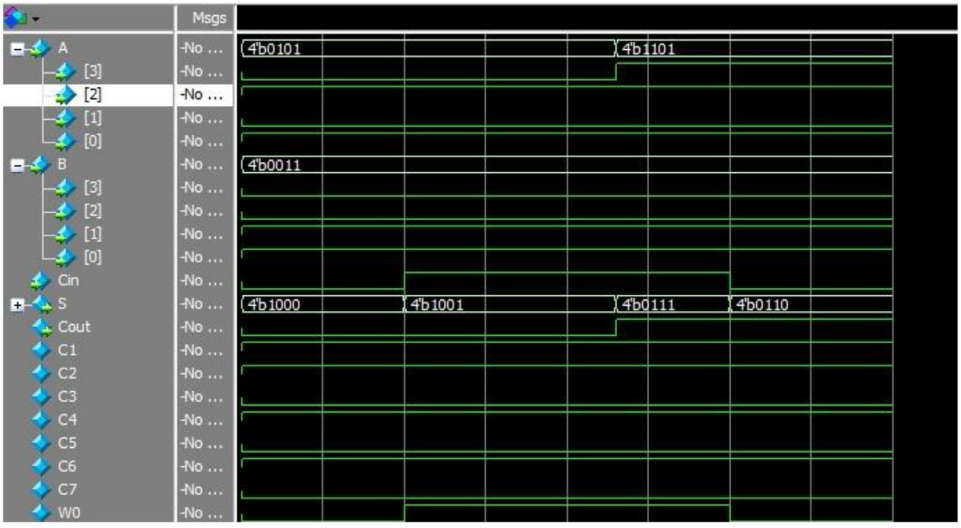


Fig. 9. BCD output for the existing system.

```
Timing Detail:
--------------
All values displayed in nanoseconds (ns)

=========================================================================
Timing constraint: Default path analysis
  Total number of paths / destination ports: 123 / 5
-------------------------------------------------------------------------
Delay:               15.582ns (Levels of Logic = 7)
  Source:            B<0> (PAD)
  Destination:       Cout (PAD)

  Data Path: B<0> to Cout
                                Gate     Net
    Cell:in->out      fanout   Delay   Delay  Logical Name (Net Name)
    ----------------------------------------  ------------
     IBUF:I->O             2    0.821   1.216  B_0_IBUF (B_0_IBUF)
     LUT3:I0->O            2    0.551   1.072  C0_block/Out1 (C0)
     LUT3:I1->O            2    0.551   1.072  C1_block/Out1 (C1)
     LUT3:I1->O            2    0.551   0.945  C2_block/Out1 (C2)
     LUT3:I2->O            4    0.551   1.256  W3_block/Q1 (W3)
     LUT4:I0->O            1    0.551   0.801  Sub_block2/Q1 (Cout_OBUF)
     OBUF:I->O                  5.644          Cout_OBUF (Cout)
    ----------------------------------------
    Total                     15.582ns (9.220ns logic, 6.362ns route)
                                       (59.2% logic, 40.8% route)
```

Fig. 10. Delay of existing method.

Device Utilization Summary				
Logic Utilization	**Used**	**Available**	**Utilization**	**Note(s)**
Number of 4 input LUTs	14	3,840	1%	
Logic Distribution				
Number of occupied Slices	8	1,920	1%	
Number of Slices containing only related logic	8	8	100%	
Number of Slices containing unrelated logic	0	8	0%	
Total Number of 4 input LUTs	14	3,840	1%	
Number of bonded IOBs	14	97	14%	
Total equivalent gate count for design	87			
Additional JTAG gate count for IOBs	672			

Fig. 11. Area of the existing system.

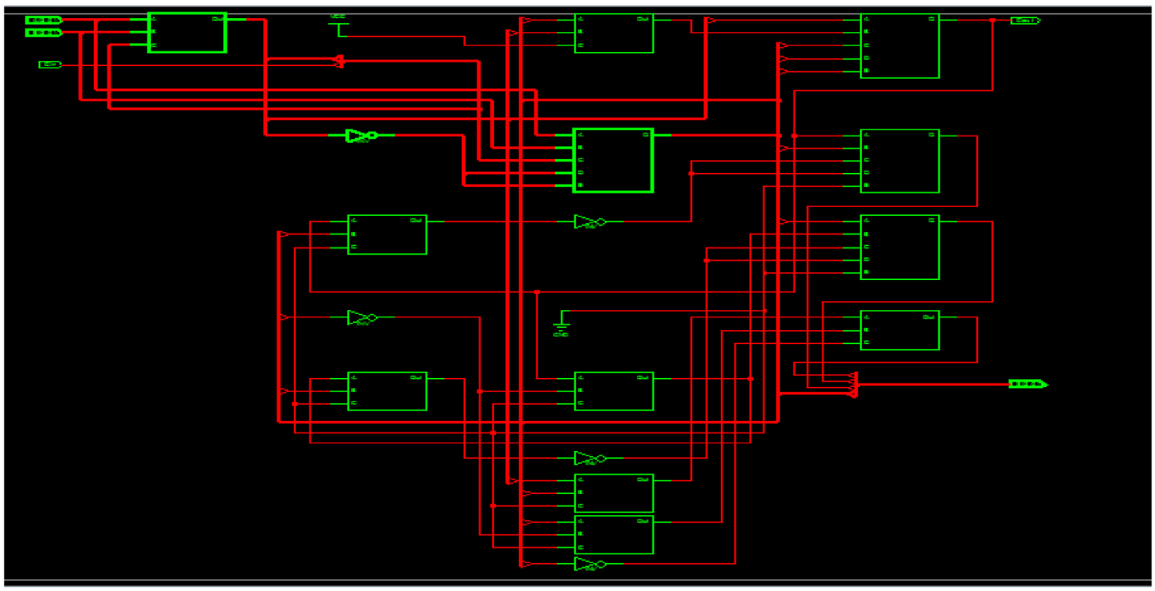


Fig. 12. RTL schematic of proposed system.

		Device Utilizat
Logic Utilization	**Used**	
Number of 4 input LUTs	12	
Logic Distribution		
Number of occupied Slices	7	
Number of Slices containing only related logic	7	
Number of Slices containing unrelated logic	0	
Total Number of 4 input LUTs	12	
Number of bonded IOBs	14	
Total equivalent gate count for design	72	
Additional JTAG gate count for IOBs	672	

Fig. 13. Area of proposed system.

adder output. Each and every bit of the input and output is important in determining the proper working of the adder.

In order to perform the XOR functionality of 2 inputs named X, Y with MGs the expression will be

$$x \oplus y = M5(x, y, z', z', 0) \tag{22}$$

where equation for Z can be given by

$$z = M3(x, y, 0) \tag{23}$$

4.4. Software requirements

4.4.1. MODELSIM

Modelsim provides simulation environment which supports multiple hardware description languages like VHDL, Verilog, System Verilog, and System C etc.... This simulation environment is created by Mentor Graphics. Verilog HDL is used to implement the simulation of majority gates.

4.4.2. Xilinx ise

Xilinx ISE provides simulation and synthesis support for researchers. It helps the explorers to adapt the intelligent and developing world. Various kits and products are provided by Xilinx which assist them in area, power and delay analysis.

5. Result and discussion

The aim of the project which was to build the BCD adder using the combination of majority gate with 5 and 3 inputs is achieved. Existing methodology deals with the BCD adder which is implemented entirely by the 3 input majority gates and inverter. The existing system uses the modified equations derived from the conventional BCD adder. In existing system with careful examinations of the DFA logical equations they achieved a reduced delay. The Fig. 9 represents the BCD output of existing method.

The RTL schematic of the existing system which shows the difference in implementation of proposed system is given in the Fig. 8 and Fig. 13. Fig. 10 represents the delay value of existing

```
Data Path: B<0> to Cout
                           Gate     Net
  Cell:in->out      fanout   Delay   Delay  Logical Name (Net Name)
  ----------------------------------------  ------------
   IBUF:I->O             2   0.821   1.216  B_0_IBUF (B_0_IBUF)
   LUT3:I0->O            3   0.551   0.907  Sub_block3/Out11 (N1)
   MUXF5:S->O            3   0.621   1.246  Sub_block9/Out1_f5 (N21)
   LUT3:I0->O            1   0.551   1.140  Cout_SW1 (N29)
   LUT4:I0->O            1   0.551   0.801  Cout (Cout_OBUF)
   OBUF:I->O                 5.644          Cout_OBUF (Cout)
  ----------------------------------------
  Total                    14.049ns (8.739ns logic, 5.310ns route)
                                    (62.2% logic, 37.8% route)
```

Fig. 14. Delay of proposed system.

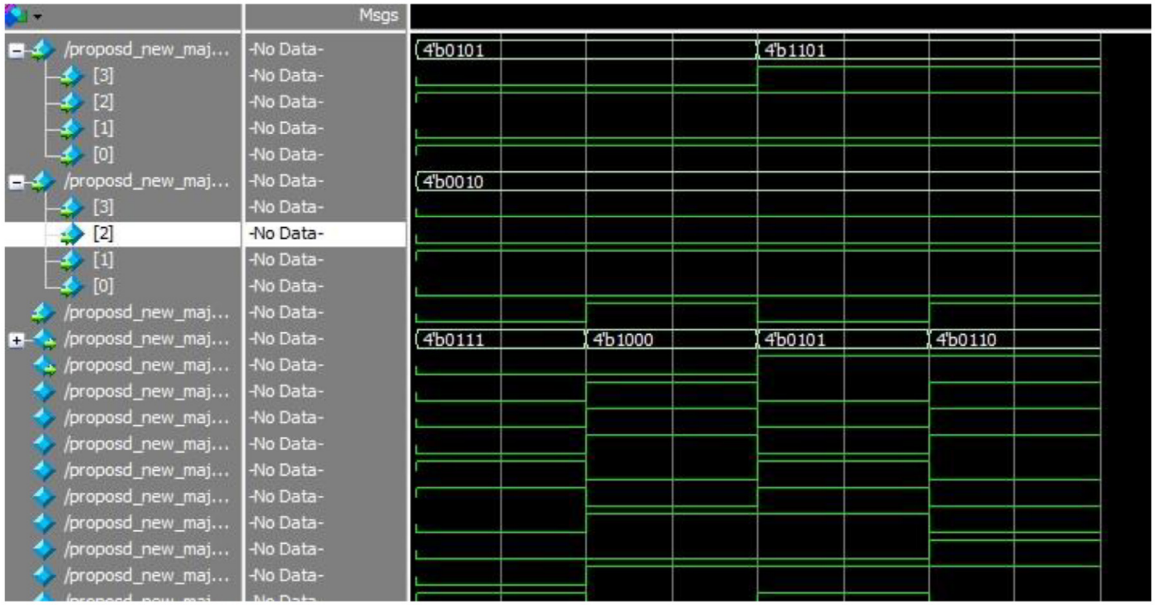


Fig. 15. Output waveform of proposed system.

system and the Fig. 11 and 14 represents the area of existing and proposed method respectively.

In proposed system the way of implementation is changed by using the 5 input majority gate. Compared to the existing system the crossover complexities is reduced without increase in area power and delay constraints. The partly used and completely used MGs are reduced compared to the existing one. The partly used and the completely used MGs were reduced from 9 to 16 to 8 and 11 respectively.

This paper deals with the 4 bit adder. So the output may vary from 0000 to 1111 with carry out 0 and 1 depending upon the input. The existing system of this paper deals with the delay of 15.582 ns and the delay of proposed system is 14.049 ns. So there is a decrease in delay which increase the speed of operation leads to the improvement of efficiency. The end product is a delay efficient binary coded decimal adder from which we can obtain a coded input for decimal input.

The Figs. 13–15 shows that the output waveform of proposed method, Xilinx synthesis area and delay constraints respectively. The RTL schematic of the proposed method are also given as follows.

	DELAY (ns)	TOTAL EQUIVALENT GATECOUNT FOR DESIGN
EXISTING	15.582	87
PROPOSED	14.049	72

6. Conclusion

MG (majority gate) is the fundamental QCA component when joins with the negator (inverter) forms the entire logical set, because the remaining logical conjunction and logical disjunction gates are defined in terms of partly used MG, where one of the input is 0 or 1(constants), respectively. In this paper, a BCD adder is introduced which is implemented with the help of majority gate with 5 input and 3 input and the negator where existing system consists of BCD implementation using inverters and majority gate with 3 inputs.

With this new type of BCD implementation using the amalgamation of majority gates with 3 and 5 inputs, crossover complexities can be reduced with reducing the delay and the count of partly as well as completely used gates. The future work of this

paper is to focus in improvement of delay, area, or power factors of BCD adder or achieving the BCD adder based on MGs with 5 inputs on QCA designer.

Ethical approval

This article does not contain any studies with human participants or animals performed by any of the authors.

Declaration of Competing Interest

This paper has not communicated anywhere till this moment, now only it is communicated to your esteemed journal for the publication with the knowledge of all co-authors.

References

[1] AngomSachindro Singh, V. Sreevani, Area-Delay efficient binary adders in QCA, International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering 3 (11) (2014).

[2] Dariush Abedi, Ghassem Jaberipur, Decimal full adders specially designed for quantum-dot cellular automata, IEEE Transactions on Circuits and System 2 (2018).

[3] Gregory L. Snider, Alexei O. Orlov, Islamshah Amlani, Gary H. Bernstein, Craig S. Lent, James L. Merz, Wolfgang porod, Quantum – Dot Cellular automata: line and majority logic gate, Jpn J Appl Phys 38 (1999) 7227–7229.

[4] Gurmohan Singh, R.K. Sarin and Balwinder Raj (2017) "A review of quantum-dot cellular automata based adders", Vol. 10, No. 4, pp. 41–58.

[5] Keivan Navi, Amir Mokhtar Chabi, Samira Sayedsalehi, A novel seven input majority gate in quantum-dot cellular automata, IJCSI International Journal of Computer Science Issues 9 (1) (2012).

[6] Keivan Navi, Razieh Farazkish, Mostafa Rahimi Azghadi, Samira Sayedsalehi, Five-Input majority gate a new device for quantum-dot cellular automata, J Comput Theor Nanosci (2010), doi:10.1166/jctn.2010.1517.

[7] Konrad Walus, Timothy J. Dysart, Graham A. Jullien, Arief R. Budiman, QCA Designer: a rapid design and simulation tool for quantum-dot cellular automata, IEEE Trans Nanotechnol (2004), doi:10.1109/TNANO.2003.820815.

[8] Maryam Taghizadeh, Mehdi Askari, Khossro Fardad, BCD computing structures in Quantum- Dot Cellular automata, in: Proceedings of the International Conference on Computer and Communication Engineering, May 13-15, 2008, Kuala Lumpur, Malaysia, 2008.

[9] Ramanand Jaiswal, Trailokya Nath Sasamal, Efficient design of full adder and subtractor using 5-input majority gate in QCA, Tenth International Conference on Contemporary Computing (IC3), 2017.

[10] Rami Akeela, Meghanad D. Wagh, A Five-Input Majority Gate in Quantum Dot Cellular Automata, 2, NSTI-Nanotech 2011, 2011 ISBN 978-1-4398-7139-3.

[11] Sang-Ho Shin, Kee-Young Yoo, Gil-Je Lee, Design of exclusive-or logic gate on quantum-dot cellular automata, International Journal of Control and Automation 8 (2) (2015) 95–104 (2015).

[12] Shoubhik Gupta, Rohit Kumar, Bahniman Ghosh, Adder design using a 5-input majority gate in a novel multilayer gate design paradigm for quantum dot cellular automata circuits, Journal of Semiconductors 36 (4) (2015).

[13] Subhashee Basu, Realization of xor and xnor gates using QCA basic gates, International Journal of VLSI and Embedded Systems-IJVES 05 (2014) 10473 ISSN: 2249 – 6556.

[14] R. Radha Krishna, K. Vijaya Krishna, Design of the best area-delay adders with QCA majority logic gates by using vhdl, International Journal of Scientific Research Engineering & Technology (IJSRET) 4 (10) (2015) ISSN 2278 – 0882.

Dr. Arulkarthick V J, received his B.E. in Electrical and Electronics Engineering from Bharathiyar University, Coimbatore; ME in Applied Electronics and Ph.D in Information and Communication Engineering from Anna University, Chennai, His research interests are Signal and Image processing, VLSI Design, and VLSI signal processing. He is currently working as Assistant Professor (Selection Grade) at JCT college of Engineering, Coimbatore. He is a member of IEEE, ISTE.

Abinaya Rathinaswamy received her M.E. degree in VLSI Design and B.E. degree in Electronics and Communication Engineering from Anna University, Chennai. Her current research interests include VLSI Design and Signal Processing. She is currently working as Assistant Professor at JCT college of Engineering, Coimbatore.

Dr. K. Srihari received the M.E. and Ph.D. degree from Anna University, Chennai. He is currently working as an Associate Professor in the Department of Computer Science and Engineering, SNS College of Engineering, affiliated to Anna University- Chennai, Tamilnadu, India. Dr. K. Srihari published over 50 papers in international journals and his research area includes semantic search engines, Microprocessor, FPGA, big data and cloud computing.