**DESIGN AND SIMULATION OF A FIVE-INPUT MAJORITY VOTER CIRCUIT USING QUANTUM-DOT CELLULAR AUTOMATA (QCA)**

J. Kirubakaran1, N Anand 2

Assistant Professor1, PG Student 2

Department of Electronics and Communication, Muthayammal Engineering college, Namakkal,

Tamil Nadu India1

Department of Electronics and Communication, Muthayammal Engineering college, Namakkal,

Tamil Nadu India2

kirubakaranj.rnd@gmail.com1, 2002anand.n@gmail.com2

**Abstract —**

**The main work of the research is Scaling of CMOS circuits has become extremely difficult at nanoscale levels. To design circuits with high packaging density, low power, and a smaller area, researchers are exploring alternative technologies. Quantum-dot Cellular Automata (QCA) is one such alternative proposed for circuit design at the nanoscale level. A voter circuit is an essential component of a redundancy-based fault-tolerant system, ensuring proper operation despite faults in system components. The robustness of the voter circuit defines the reliability of the fault-tolerant system. This work presents simulation results and analysis of a QCA-based fault-tolerant five-input majority voter circuit. In a Triple Modular Redundant (TMR) system, the robustness of the voter circuit has been enhanced by incorporating a five-input majority gate. The proposed QCA cell can also be implemented in a single layer, reducing complexity. Simulations of the QCA cell are performed based on analytical calculations, demonstrating its efficiency and reliability.**

***Keywords*- CMOS; QCA; TMR; LSI; ASIC.**

1. **INTRODUCTION**

Since the electronics technology accomplished higher levels of integration into a single Silicon chip that led to Large Scale Integration (LSI), which preceded VLSI, the applications of electronic systems have experienced an almost unlimited expansion. However, despite the many advantages provided by VLSI, the inherent high integration level started to necessitate very sophisticated testing strategies in order to verify the correct device operation. As the electronics market stimulated the use of VLSI in a variety of tasks from critical military applications to consumer products, the reliability of the products’ functioning gained an escalating importance.

The expanding demand for ASIC applications led to development of more sophisticated Computer Aided Design (CAD) tools; which have shown most significant progress in layout and simulation, with yet more inferior improvement in testing. This consequently leads to designs with superior complexity, but which are in contrast extremely difficult to test effectively. Moreover, due to the low volume attribute of ASICs, the high-test costs cannot be retaliated with large amounts of mass production. In the early times of electronics engineering, when systems were constituted from discrete components, testing of digital systems comprised three distinct phases:

1- Each discrete component was tested for concordance to its specifications

2- The components were assembled into more complex digital elements (i.e. flip flops etc.), and these were tested for correct functionality Testing and Built in Self Test

3- The higher-level system was built up and was tested for functionality

As the systems acquired higher complexity, the 3rd phase began to become increasingly difficult to accomplish and other means of system verification were begun to be sought. In R.D. Eldred suggested another way, which is well-known and used as structural test at present, in order to test the hardware of a system instead of the burdensome functional test.

The first applications of the proposed structural test was to discrete components on Printed Circuit Boards (PCBs), which then began to be applied to ICs as the electronics technology developed into higher levels of integration.Though the problems of IC testing was not very much different from that of the PCBs, the objective of testing had then changed to discard the faulty units rather than locating the defective components and replace them. In the case of Small-Scale Integration (SSI) and Medium Scale Integration (MSI), the problems were relatively as simple as PCB testing, since:

1- Internal nodes of the devices were easily controlled and observed from the primary inputs and outputs of the devices.

2- The simplicity of circuit functions permitted the use of exhaustive testing

3- More complex systems were constructed from basic, thoroughly tested components.

* 1. **FUNCTIONAL AND STRUCTURAL TESTING**

Before structural testing was proposed, digital systems were tested to verify their compliance with their intended functionality, i.e. in this philosophy, a multiplier would be tested whether it would multiply and so forth. This testing philosophy is termed as functional testing, which can be defined as, applying a series of determined meaningful inputs to check for the correct output responses in terms of the device functionality. Although this methodology imparts a good notion of circuit functionality, under the presence of a definitive fault model, it is very difficult to isolate certain faults in the circuits in order to verify their detection. With the proposal of structural testing, which might be defined as, consideration of possible faults that may occur in a digital circuit and applying a set of inputs tailored for detecting these specified faults. . As obvious structural testing relies on the fault Testing and Built in Self Test models described for the Device Under Test (DUT), and any result obtained in this manner is unworthy without a proper description of used fault models. Fault models and fault simulation techniques developed as a result of the above described technique will be elaborated.

* 1. **NEED FOR TESTING**

Circuit manufacturers must thoroughly test their products before deliv­ering them to customers. The causes of circuit failure can be divided into two main cate­gories: design errors and manufacturing defects.

Design errors are caused by errors in the layout. If the errors can be eliminated by chang­ing the layout, then it is considered a design error. We normally attempt to detect design errors by simulating the circuit and testing the simulation. No simulation can perfectly predict what a real circuit will do. For example, most simulations cannot predict whether or not latch up will occur. Thus, it is necessary to test a real circuit before we can be sure that all design errors have been fixed.



**Figure 1: chip area vs yield**

Manufacturing defects are caused by random variations in the manufacturing process that can cause malfunctions in circuit components. These defects can occur even in circuits that have no design errors.Most modern processes have an average defect density of around 1 defect per cm2. As the chip area increases, the probability of a defective chip increases and the yield (probability of no defects) decreases .Chip is yield for performance reasons, we would like to have the chip as large as possible. This means that a considerable fraction of large chips will be defective and we must test each chip to be sure that it is defect. Before proceeding further, let us be clear on what we mean by testing a circuit. Suppose we have a digital circuit which we wish to test. We will give it a set of inputs (usually called “test vectors”) and we will observe the outputs to see if they are correct.



**Figure 2 : CUT**

The correctness is done by

1.Is the output signal equivalent to the correct logic value, i.e. is it functionally correct?

2.Does the output reach its correct value in a timely fashion, i.e. is it fast enough?

3.Does the circuit consume too much power?

There might be other test criterion as well, such as susceptibility to temperature variations, radiation or mechanical vibration, but we will not consider them here.

* 1. **BACKGROUND OF QCA**

In 1993, Craig Lent proposed a new concept called quantum-dot cellular automata (QCA). This emerging technology has made a direct deviation to replace conventional CMOS technology based on silicon. QCA generally uses arrays of coupled quantum dots in order to implement different Boolean logic functions. QCA or quantum-dot cellular automata as its name is pronounced uses the quantum mechanical phenomena for the physical implementation of cellular automata. In the general case, conventional digital technologies require a range of voltages or currents to have logical values, whereas in QCA technology, the position of the electrons gives an idea of the binary values. The advantages of this technology are especially given in terms of speed (range of terahertz), density (50 Gbits/cm2) and in terms of energy or power dissipation (100 W/cm2).

QCA is based essentially on a cell. Each cell represents a bit by a suitable charge configuration as shown in Figure 3. It consists of four quantum dots and two electrons charge. Under the effect of the force of Colombian repulsion, the two electrons can be placed only in two quantum sites diametrically opposite.

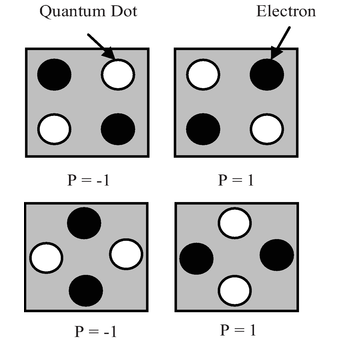


Figure 3 . Basic QCA cell**.**

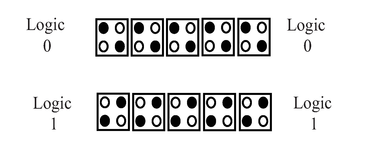
A QCA cell is composed of four points with one electron each in two of the four points occupying diametrically opposite locations. The question that arises in this case is why do electrons occupy quantum dots of opposite or diagonal corner

To answer this question, it is enough to have an idea about the principle of the repulsion of Coulomb, which is less effective with respect to the electrons when they are in adjacent quantum dots. The points are coupled to one another by tunnel junctions.

Thus, the internal effect of the cell highlights two configurations possible; each one will be used to represent a binary state “0” or “1.” A topology of QCA is a paving of cells QCA. The interaction between the cells makes it possible to transmit information which gives the possibility of replacing physical interconnection of the devices.

The information (logic 0 or logic 1) can propagate from input to the output of the QCA cell only by taking advantage of the force of repulsion as shown in Figure 3. Clocking is an important term in QCA design. In order to propagate the information through QCA without any random adjustments of QCA cells,

The QCA is necessary to make a clock to guarantee the same data putting from input to the output. According to  timing in QCA is obtained by clocking in four distinct periodic phases namely Switch, Release, Relax, and Hold.

 Figure 4.Operations of a QCA wire propagation by application of logic 0 and logic 1 1 to a QCA cell at the input.

Many architectures of logic devices can be designed by using adequate arranging of QCA cells. The biggest advantage of this wireless technology is that the logic is carried by the cells themselves. The fundamental QCA logic is binary wire, inverter, and majority voter. These QCA logic gates are evaluated and simulated using the QCA Designer tool version Noninverted gate or binary wire the great advantage of cell QCA is that all the close cells are aligned on a specific polarization, which depends on the input cell or the driver cell. Hence, by arranging the cells side-by-side according to the type “0” or “1” applied to the input cell, any logic can be transferred. Consequently, this gate can play the role of a wire or binary interconnection or noninverted gate The layout of each cell given by binary wire is represented in Figure 5.

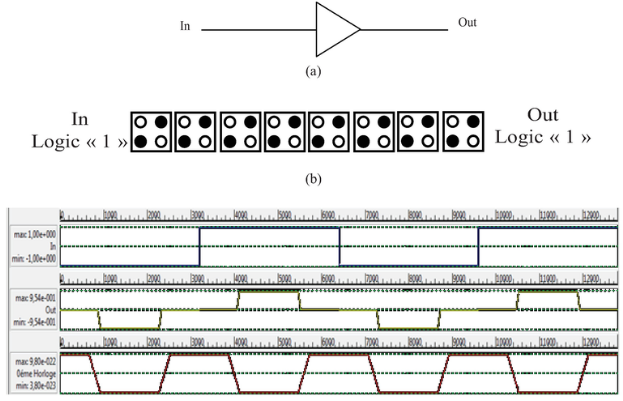


Figure 5 Binary wire, (a) representation, (b) QCA layout,

The simulation results of the noninverter gate are presented in . One waveform with one frequency is applied to the input (In), one waveform for the first clock (Clk 0), and one waveform for the binary wire outputs (Out). From simulation results of binary wire given by , the expression from the output pulses of the noninverter gate can be deduced, which is given by [Eq. (1)](https://www.intechopen.com/books/advanced-electronic-circuits-principles-architectures-and-applications-on-emerging-technologies/nanoarchitecture-of-quantum-dot-cellular-automata-qca-using-small-area-for-digital-circuits" \l "E1):

Out = In . Clk¯ 0 (1)

Thanks to the columbic interaction between electrons in neighboring cells, different types of the inverter gates in QCA were proposed. The principal operation of this gate is to invert the input signal applied. If the applied input is low then the output becomes high and vice versa, The input “In” is given to one of the ends and the output reversed will be obtained at the output “Out.” The position of the electrons and the layout of each cell are represented. The principle of operation of this gate is based on the wire of input, which will be prolonged in two parallel wires and will polarize the cell placed at the end of these two wires, which implies the opposite polarization of this cell due to the Coulomb repulsion.

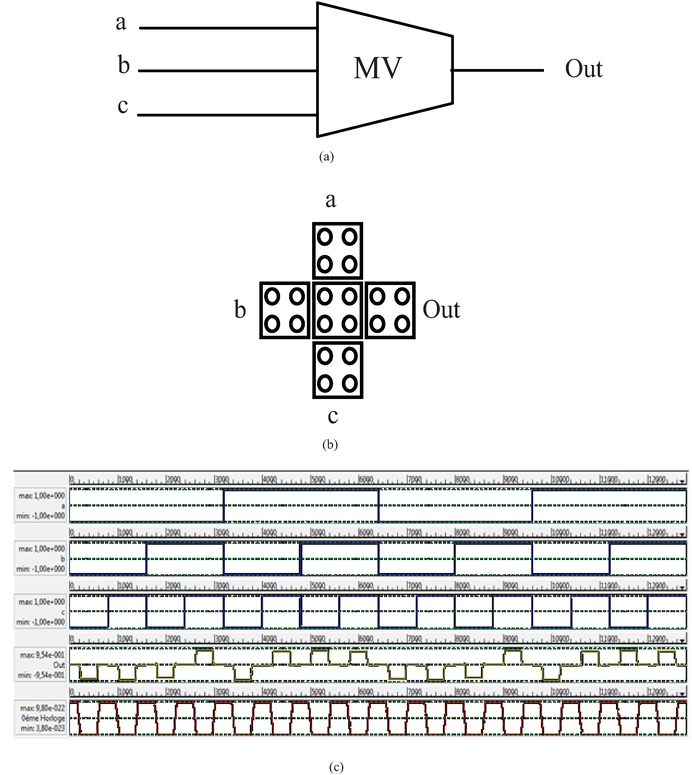


Figure 6 Majority voter (MV) gate, (a) QCA representation, (b) QCA layout, and (c) simulation results.

In conventional digital VLSI design, it is assumed that a circuit/system should function perfectly to provide accurate results. In non-digital world, ideal operations are seldom needed e.g. “analog computation” that provides “good enough” results instead of totally accurate results may in fact be acceptable. For many digital systems, the data already contained errors e.g. in a communication system error may occur everywhere. In communication system, the analog signal out coming from the outer world must first be sampled before being transformed to digital data at the frontend of the system. The digital data are then processed and transmitted in a noisy channel before convert back to an analog signal at the back end of the system. Errors may take place anywhere during this process. Due to the advance in transistor size scaling, factors such as noise and process variations which are previously not important are becoming important in today’s digital IC design.

**II. EXISTING SYSTEM**

**A Self-Checking TMR Voter for Increased Reliability Consensus Voting in FPGAs**

Umar Afzaal analyzed that Voting circuits used in TMR implementations are decentralized and consensus is calculated from the redundant outputs off-chip. However, if there is an insufficient number of pins available on the chip carrier, the TMR system must be reduced to an on chip unprotected simplex system, meaning voters used at those locations become single points of failure. This literature proposes a self-checking voting circuit for increased reliability consensus voting on FPGAs.



Fig. 1. Logic design of the voter

The logic design of the proposed self checking voter. The voter achieves fault-tolerance by means of shifting between redundant voter copies. There are four copies of a majority voting block that are grouped into two sets: set-A and set-B. During operation, only one of the two sets feeds lines V1 and V2. Voter output is taken from line V1 in Fig. 3, but we can select either of the lines V1 or V2 as the output as both of them have identical signal assertions. Each copy from set-A is matched against a copy from set-B using two flags called conflict flags: CF1 and CF2. An additional logic unit called the arbiter monitors V1 and V2 to control the selection between the two sets of voters by driving CF1 and CF2 accordingly. If only, a single flag was used, then the voter would only be able to shift once to the redundant voter set. However, the use of two flags by the arbiter allows flip-flop shifting between the two voter sets.

In order to understand the operation of the circuit, let us assume that set-A is currently active and sourcing the lines V1 and V2 (i.e., the value of CF1 is set to ‘0’ at this point to select set-A). Additionally, let us assume that the current inputs to the voter are ('M1,M2,M3'= '0,0,0') such that the voter produces logic-‘0’ on lines V1 and V2. Here M1, M2, and M3 are the outputs of the three TMR modules. When both V1 and V2 are ‘0’ and CF1 is also ‘0’, mux-x outputs ‘1’ which selects CF1 at mux-z to set a ‘0‘ value for CF2.

Thus, the signal states at this point are ('V1,V2,CF1,CF2'= '0,0,0,0'). This condition is represented by state-1 in Fig. Now, assume that an upset introduces a fault in one of the two copies belonging to set-A and that this fault presents as an error in the input vector ('M1,M2,M3'= '0,0,0'). If this error presents, the arbiter observes a difference between V1 and V2 such that ('V1,V2,CF1,CF2'= '1,0,0,0'). In this case, the arbiter complements CF1 to switch to set-B ('V1,V2,CF1,CF2'= '1,0,1,0'). This is the transition to state-2. Assuming that set-B is fault-free, lines V1 and V2 will agree with each other again, which is then observed by the arbiter as ('V1,V2,CF1,CF2'= '0,0,1,0'). The arbiter then changes the value of CF2 to match that of CF1, which is represented by the transition back to state-1 ('V1,V2,CF1,CF2'= '0,0,1,1'). At this point, the voter can still tolerate another fault that does not overlap with the existing fault (i.e for some other combination of 'M1,M2,M3'). Given a repair mechanism such as CRAM scrubbing is active to correct the existing fault, it is also possible to tolerate overlapping non-concurrent faults

.

**III PROPOSED SYSTEM**

Nano electronic systems are now more and more prone to faults and defects, permanent or transient. Redundancy techniques are implemented widely to increase the reliability, especially the TMR - Triple Modular Redundancy. However, many researchers assume that the voter is perfect and this may not be true.

This work proposes a simple but effective fault tolerant voter circuit which is more reliable and less expensive. On the right, the five-redundant voter expands the redundancy by incorporating five voter modules. This design offers superior fault tolerance by utilizing additional modules to handle potential errors more effectively. Inputs (Gin and S) are distributed across the five modules, and the outputs (Cout and Out) are derived after redundancy checks among all modules. While the five-redundant voter enhances reliability, it comes with increased complexity and requires more QCA cells than the three-redundant voter.



. Figure 1. proposed voter

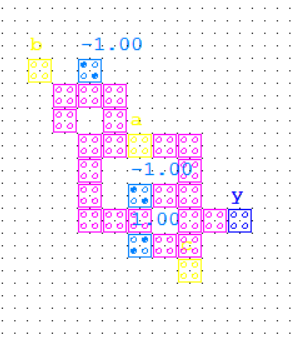
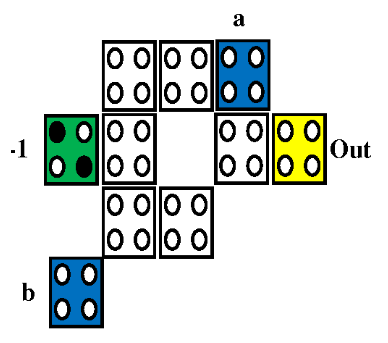


Figure 2 proposed QCA voter

In this work we propose a new scheme for majority voting presented in figure . This circuit is based on few logic blocks, an exclusive or gate and a multiplexer. The structure of XOR gate and MUX used in this work are shown in Figure



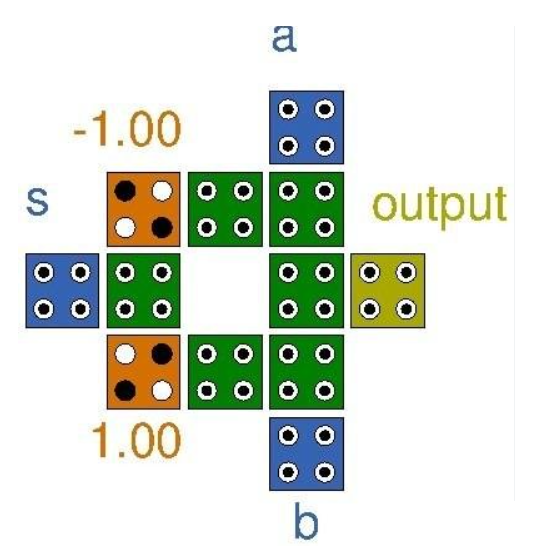


Fig 3. Output

The proposed voter structure shows a better performance than the previous schemes. Compared to Kshirsagar’s voter, obviously it consumes less area overhead together with power dissipation and also there is less delay (and consequently also better than the classical structure), and these parameters are all important characteristics for a voter. These features and improvements come directly from the simplicity of the architecture. Further , the voter with different redundancy techniques are implemented with three and five majority voters.

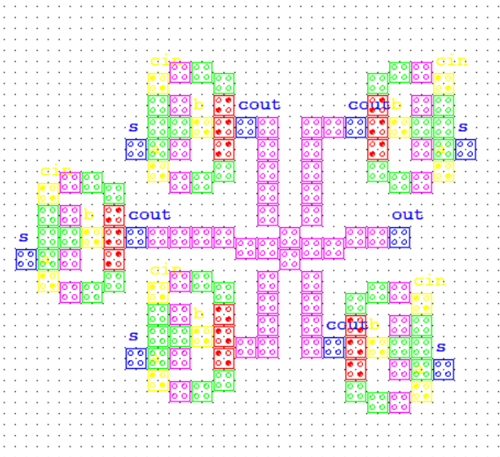


Fig.4 Five redundant voters

The designs of three-redundant and five-redundant voter structures implemented in QCA technology. These voter designs are essential for achieving fault tolerance in digital circuits, ensuring reliable operation even in the presence of errors. The three-redundant voter on the left consists of three interconnected voter modules that process the input signals (Gin and S) to generate fault-tolerant outputs (Cout and Out). This design minimizes the number of QCA cells, making it compact and energy-efficient, while still providing a basic level of error correction.

On the right, the five-redundant voter expands the redundancy by incorporating five voter modules. This design offers superior fault tolerance by utilizing additional modules to handle potential errors more effectively. Inputs (Gin and S) are distributed across the five modules, and the outputs (Cout and Out) are derived after redundancy checks among all modules. While the five-redundant voter enhances reliability, it comes with increased complexity and requires more QCA cells than the three-redundant voter.

Both designs showcase scalable fault-tolerant logic in QCA, with the three-redundant voter being more suitable for applications prioritizing compactness and efficiency, and the five-redundant voter being ideal for critical applications requiring high reliability. These designs highlight the adaptability of QCA technology for building robust and efficient nanocomputing systems.

**IV RESULTS AND DISCUSSION**

QCA Designer tool is a simulation software which provides a very fast and exact simulation of digital circuits using quantum-dot cellular automata (QCA) in Nanotechnology. Also it is preferred for simulating any complex QCA circuits on most standard stages. This simulator was at first planned in ATIPS lab, University of Calgary, Calgary, Canada.

The QCA Designer has attracted some significant new engineers, including top scientists from the University of Notre Dame. The undertaking is written in C/C++ and utilizes a wide scope of open-source programming, for example, GNU image manipulation program toolkit (GTK+) designs library, and it is kept up under the GNU's not UNIX (GNU) public permit (GPL) for open-source programming. The goal of the undertaking is to make simple to access simulation and design instrument open to all examination network through the internet. Different designers ought to have the option to effortlessly incorporate their own utilities into QCA Designer. This is developed by giving normalized technique to represent the data inside the software. Also, simulation engines can be incorporated into a QCA Designer tool, utilizing a normalized model and data types without much of a stretch.

The recent version of QCA Designer incorporates the simulation engines in three different types. The first is a logic simulator in digital form, which beliefs cells to be either invalid or completely polarized. The second is a nonlinear approximation engine that utilizes a nonlinear cell to cell reaction to iteratively decide the steady condition of the cells inside a model.

The third uses a two-state Hamiltonian to frame a full quantum mechanical model of approximation for such a framework. One of the primary issues in executing more precise simulations are the absence of experimental information for QCA frameworks with countless cells. Subsequently, the target of this exertion is to give inspiration to additional examination for developing a discourse between circuit designers and implementers. Future adaptations in QCA Designer will incorporate simulation engines based on the outcome acquired from experimentation. A QCA Designer has three simulation engines. Each engine has an alternate and significant arrangement benefits and disadvantages. The simulation engine can play out a comprehensive confirmation of the framework or a set of client selected vectors.

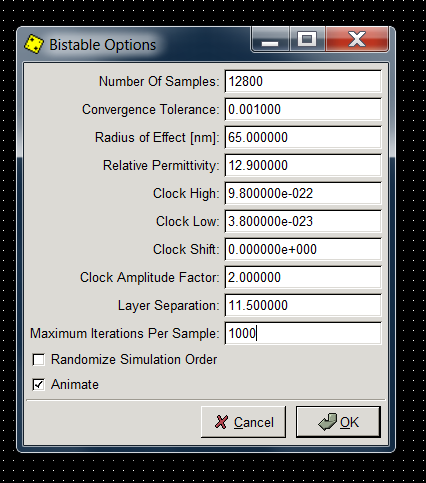


Fig1. Input Sequence & Output Sequence of configurable cell

****

Fig. 2. QCA Layout of proposed voter

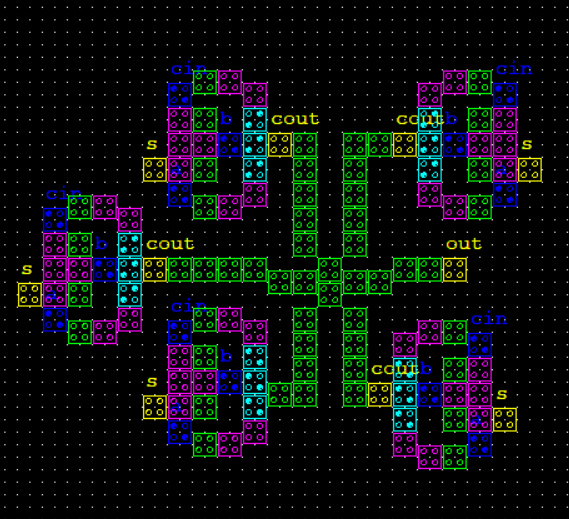


Figure 3 QCA Layout of proposed three and five voter

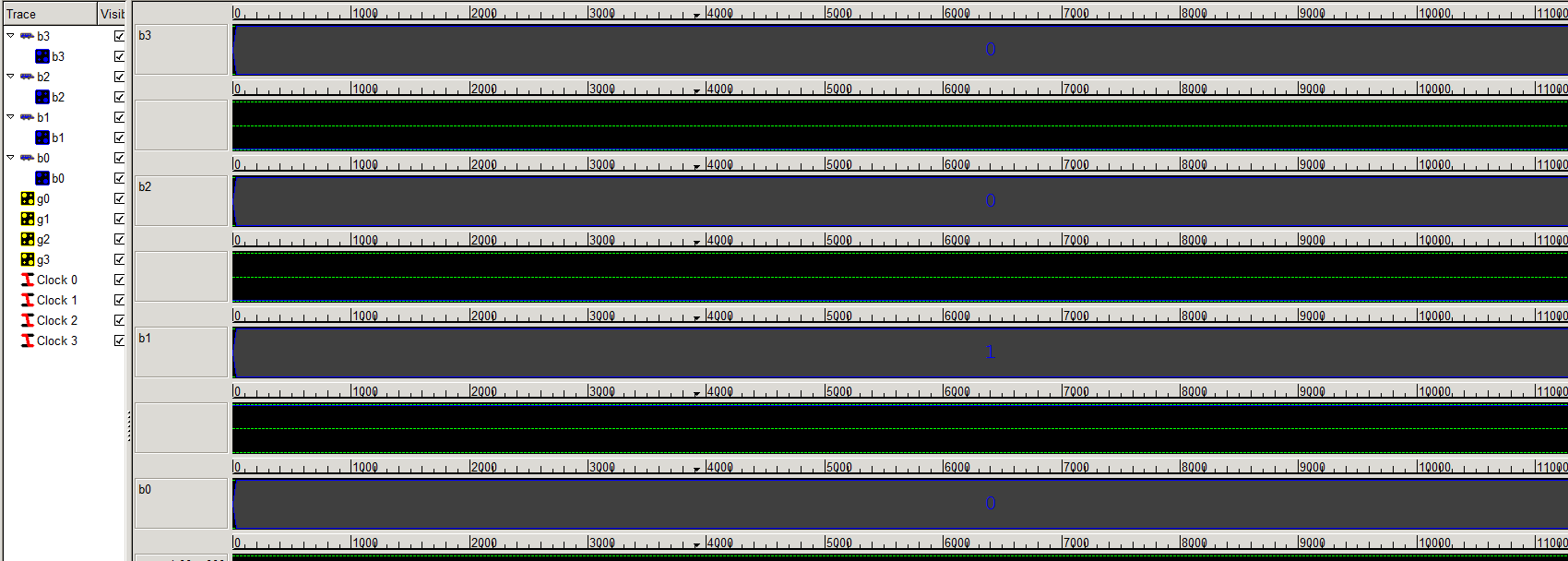


Fig 4. Input sequence

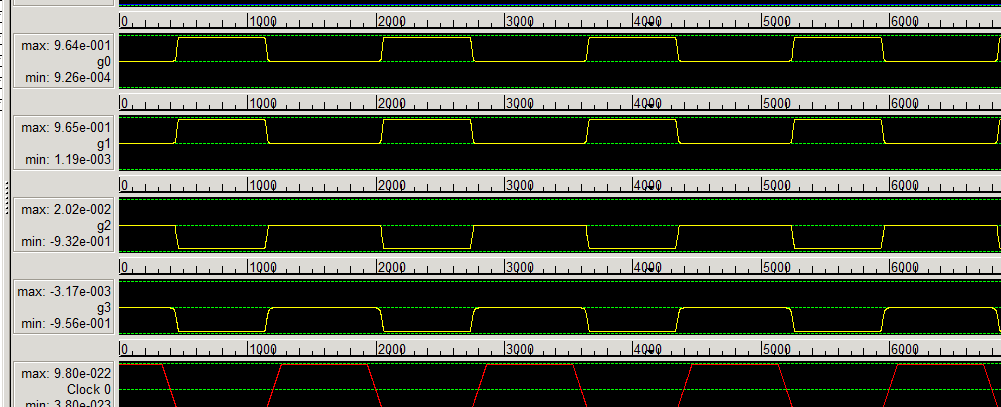


Figure 5 output sequence

By comparing the output waveforms with the intended logic, users can ensure that the circuit performs as expected. Any discrepancies in the waveforms indicate potential design flaws, such as incorrect cell arrangements or clocking issues.

The functional waveform verification ensures that the proposed voter design meets the desired specifications before moving to physical implementation.

|  |  |  |
| --- | --- | --- |
| **Author** | **Dots** | **Delay clock count** |
| D. Abedi | **54** | **4** |
| M. Pathania | **48** | **4** |
| M. Javid | **48** | **3** |
| D. Tripathi | **36** | **3** |
| T. Sultana | **33** | **3** |
| **Proposed** | **28** | **1** |

Table . Comparison table

The table presents a comparative analysis of the proposed QCA voter design with existing designs in terms of key parameters: dots (number of QCA cells) and delay clock count (the number of clock cycles required for operation).

**V CONCLUSION**

This work presented a QCA based digital voter circuit to be used in TMR schemes. This novel approach has a simple architecture, saves area, has a low power consumption and less propagation delays. It is also a robust single fault solution that exceeds the reliability of current solutions. Future work could expand to how to formulate an algorithm to design a voter circuit for NMR systems extends beyond TMR which has the same property as this.

**REFERENCES**

1. D. Abedi and G. Jaberipur, "Coplanar QCA serial adder and multiplier via clock-zone based crossover," *2015 18th CSI International Symposium on Computer Architecture and Digital Systems (CADS)*, Tehran, Iran, 2015, pp. 1-4, doi: 10.1109/CADS.2015.7377791.
2. M. Pathania, C. P. Singh, L. Sharma, H. Faridi and S. Gargrish, "A New XOR gate based Half Adder design in QCA Nanotechnology," *2022 2nd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE)*, Greater Noida, India, 2022, pp. 223-225, doi: 10.1109/ICACITE53722.2022.9823419.
3. M. Javid and K. Mohamadi, "Characterization and Tolerance of QCA Full Adder under Missing Cells Defects," *2009 Fifth International Conference on MEMS NANO, and Smart Systems*, Dubai, United Arab Emirates, 2009, pp. 85-88, doi: 10.1109/ICMENS.2009.22.
4. D. Tripathi and S. Wairya, "An Energy Dissipation and Cost Optimization of QCA Ripple Carry Adder," *2021 8th International Conference on Signal Processing and Integrated Networks (SPIN)*, Noida, India, 2021, pp. 760-765, doi: 10.1109/SPIN52536.2021.9566068.
5. T. Sultana, R. Bardhan, T. F. Bithee, Z. Tabassum and N. J. Lisa, "A compact design of n-bit ripple carry adder circuit using QCA architecture," *2015 IEEE/ACIS 14th International Conference on Computer and Information Science (ICIS)*, Las Vegas, NV, USA, 2015, pp. 155-160, doi: 10.1109/ICIS.2015.7166586.
6. Jing Huang, M. Momenzadeh and F. Lombardi, "Defect Tolerance of QCA Tiles," *Proceedings of the Design Automation & Test in Europe Conference*, Munich, Germany, 2006, pp. 1-6, doi: 10.1109/DATE.2006.244118.
7. ánchez-Macián, A. Martín-Toledano, J. A. Bravo-Montes, F. García-Herrero and J. A. Maestro, "Reducing the Impact of Defects in Quantum-Dot Cellular Automata (QCA) Approximate Adders at Nano Scale," in *IEEE Transactions on Emerging Topics in Computing*, vol. 10, no. 2, pp. 635-647, 1 April-June 2022, doi: 10.1109/TETC.2021.3136204.
8. Roohi, R. Zand, S. Angizi and R. F. DeMara, "A Parity-Preserving Reversible QCA Gate with Self-Checking Cascadable Resiliency," in *IEEE Transactions on Emerging Topics in Computing*, vol. 6, no. 4, pp. 450-459, 1 Oct.-Dec. 2018, doi: 10.1109/TETC.2016.2593634.
9. M. Raj, K. R. Sekar and G. Lakshminarayanan, "Majority-Logic-Based Self-Checking Adder in Quantum-Dot Cellular Automata," in *IEEE Design & Test*, vol. 39, no. 5, pp. 88-97, Oct. 2022, doi: 10.1109/MDAT.2022.3180976.
10. Z. D. Patitz, N. Park, Minsu Choi and F. J. Meyer, "QCA-based majority gate design under radius of effect-induced faults," *20th IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems (DFT'05)*, Monterey, CA, USA, 2005, pp. 217-225, doi: 10.1109/DFTVS.2005.55.
11. Sen, M. Dutta, D. Banik, D. K. Singh and B. K. Sikdar, "Design of Fault Tolerant Reversible Arithmetic Logic Unit in QCA," *2012 International Symposium on Electronic System Design (ISED)*, Kolkata, India, 2012, pp. 241-245, doi: 10.1109/ISED.2012.50.
12. M. S. Latha Gade and S. Rooban, "An Efficient Design of Fault Tolerant Reversible Multiplexer Using QCA Technology," *2020 3rd International Conference on Intelligent Sustainable Systems (ICISS)*, Thoothukudi, India, 2020, pp. 1274-1280, doi: 10.1109/ICISS49785.2020.9315867.
13. Bhowmik, A. K. Saha, P. Dutta and S. Dhar, "A Novel Design to Obtain Fault Tolerant Majority Gate for Five Input Majority Gate by Quantum Cellular Automata," *2016 International Conference on Micro-Electronics and Telecommunication Engineering (ICMETE)*, Ghaziabad, India, 2016, pp. 429-433, doi: 10.1109/ICMETE.2016.114.
14. B. Sen, R. Mukherjee, R. K. Nath and B. K. Sikdar, "Design of Fault Tolerant Universal Logic in QCA," *2014 Fifth International Symposium on Electronic System Design*, Surathkal, India, 2014, pp. 166-170, doi: 10.1109/ISED.2014.41.
15. M. Maity, P. Ghosal and B. Das, "Design of low power fault tolerant reversible multiplexer using QCA," *2012 Third International Conference on Emerging Applications of Information Technology*, Kolkata, India, 2012, pp. 467-470, doi: 10.1109/EAIT.2012.6408019.