Addressing Nano-Technology through Synthesis of Reversible Logic using Quantum-Dot Cellular Automata

Thesis submitted by

Ayan Chaudhuri

Doctor of Philosophy (Engineering)

Department of Computer Science & Engineering
Faculty Council of Engineering and Technology
Jadavpur University
Kolkata, India

2019

JADAVPUR UNIVERSITY

KOLKATA – 700032, INDIA

INDEX NO. 134/17/E

Title of Thesis

Addressing Nano-Technology through Synthesis of Reversible Logic using Quantum-Dot Cellular Automata

Name, Designation & Institution of Jt. Supervisor:

Dr. Chitrita Chaudhuri
Associate Professor,
Department of Computer Science and Engineering,
Jadavpur University, Kolkata – 700032, India

Name, Designation & Institution of Jt. Supervisor:

Dr. Diganta Sengupta
Associate Professor,
Department of Computer Science and Engineering,
Techno International Batanagar, Kolkata – 700141, India

List of Publications

International Journals:

1 Ayan Chaudhuri, Mahamuda Sultana, Diganta Sengupta, Chitrita Chaudhuri and Atal Chaudhuri, "A Reversible Approach to Two's Complement Addition using a Novel Reversible TCG Gate and its 4 Dot 2 Electron QCA Architecture", Microsystem Technologies, Springer, (25), pp 1965-1975, 2018. (SCI Indexed – Impact Factor 1.581).

2 Ayan Chaudhuri, Diganta Sengupta, Chitrita Chaudhuri, "Adiabatic Hamming Code Generator-Checker using 4-Dot-2-Electron Quantum Dot Cellular Automata", International Journal of Engineering and Advanced Technology, vol. 8, issue 6, August 2019. (SCOPUS Indexed).

International Conferences:

3 Ayan Chaudhuri, Mahamuda Sultana, Diganta Sengupta, Chitrita Chaudhuri and Atal Chaudhuri "A Comparative Analysis of CNF and XOR-AND Representations for QCA Majority Gate Estimation", *52nd Annual Convention of Computer Society of India (CSI 2017),* January 19-21, 2018, Science City, Kolkata, Springer Nature CCIS Series, vol 836, pp 402-415. (SCOPUS)

4 Ayan Chaudhuri, Mahamuda Sultana, Diganta Sengupta, and Atal Chaudhuri, "A novel reversible two's complement gate (TCG) and its quantum mapping," in *2017 Devices for Integrated Circuit (DevIC), IEEE*, Kolkata, 2017, pp. 252-256. (SCOPUS)

CERTIFICATE FROM THE SUPERVISOR

This is to certify that the thesis entitled **"Addressing Nano-Technology through Synthesis of Reversible Logic using Quantum-Dot Cellular Automata"** submitted by **Mr. Ayan Chaudhuri**, who got his name registered on 27th March, 2017, for the award of PhD (Engg.) degree of Jadavpur University is absolutely based upon his own work under the supervision of **Dr. Chitrita Chaudhuri and Dr. Diganta Sengupta** and that neither his thesis nor any part of the thesis has been submitted for any degree or any other academic award anywhere before.

SIGNATURE OF JOINT SUPERVISORS WITH DATE AND OFFICIAL SEAL

Name, Designation & Institution of Jt. Supervisor:

Dr. Chitrita Chaudhuri
Associate Professor,
Department of Computer Science and Engineering,
Jadavpur University, Kolkata – 700032, India

Name, Designation & Institution of Jt. Supervisor:

Dr. Diganta Sengupta
Associate Professor,
Department of Computer Science and Engineering,
Techno International Batanagar, Kolkata – 700141, India

DECLARATION

I hereby declare that the work described in this thesis is entirely my own. No portion of the work referred to in this thesis has been submitted in support of an application for another degree or qualification of this or any other university or institute. Any help or source information, which has been availed in the thesis, has been duly acknowledged.

Signature of the Candidate

AYAN CHAUDHURI

DATE:

Dedicated

to

My Parents

ACKNOWLEDGEMENT

This dissertation is not only the signature of my research or my skills at keyboard but resembles a milestone on my attachment with Jadavpur University; more precisely with the Department of Computer Science and Engineering. I have really been encouraged to find the “me” in myself once I have started my journey of research in the campus of wisdom, Jadavpur University. Here everything goes smooth and one starts getting the feel that nothing is impossible in this world. Obviously the thesis is the epicenter of all experiences that I have gained in the university and from a good number of remarkable individuals in my life.

First and foremost, I would like to thank my research advisor, Dr. Chitrita Chaudhuri, Associate Professor in the Department of Computer Science and Engineering, Jadavpur University. From the initial days, she has been tremendously supportive in my entire endeavor for a research topic. She never said ‘no’ in my amazing journey of research, right from the selection of the research topic to the completion of this thesis and also in between. She instilled high confidence on me and motivated me throughout my research work. She was not only my research advisor, but also my source of inspiration. In many occasions she was my strength whenever I had gone through any crisis. I feel proud to say that she was not only my research advisor rather my guide, friend and philosopher. I have no reservation to say without her academic as well as emotional assistance this work can not reach to this state.

At the same breadth I would like to acknowledge the contribution of my joint research advisor, Dr. Diganta Sengupta, Associate Professor in the Department of Computer Science and Engineering, Techno International Batanagar. He is more as my elder brother than my guide as a senior of mine from the same department. He possesses an unbelievable personality which cannot be described rather to feel and it is a great miss for those who have not the chance of coming close to him. I repeat he is more than my research guide and I personally feel blessed for his association.

I would like to express my gratitude towards Mr. Goutam Roy Chowdhury, Chairman, Techno India Group and Mr. Anit Adhikari, Director, Techno India Group for their constant inspiration, cooperation and support in carrying my research work. I also like to thank my colleagues at Techno India, Saltlake for their support as well. It would me wrong from my part if I do not mention and acknowledge the contribution of my other research mates like Mrs. Mahamuda Sultana, Mr. Prabir Kumar Naskar and Mr. Hari Narayan Khan. I must also acknowledge the sacrifice of my wife who made me free from all family responsibilities in order to complete the work in time.

I would also like to express my sincere gratitude to my thesis committee, Prof. (Dr.) Mahantapas Kundu, Head, Dept. of CSE, Jadavpur University, Prof. (Dr.) Mita Nasipuri, member, Doctorate

Committee, Jadavpur University, Prof. (Dr.) Susmita Ghosh, Prof. (Dr.) Sarmistha Neogy, Prof. (Dr.) Diganta Saha, and Prof. (Dr.) Sankhayan Choudhury, University of Calcutta for their valuable support in my endeavor.

Finally I am thankful to my parents Prof. Atal Chaudhuri and Mrs. Jhumjhum Chaudhuri for believing me, educating me and providing me with all amenities to grow up as an educated responsible citizen of my country. I still remember my grandfather and grandmother Prof. Sailes Bhushan Chaudhuri and Mrs. Bela Chaudhuri who were fundamental source of inspiration right from my childhood. It is the blessing of God that my grooming during school days was accomplished from my grandfather the great teacher. I have witnessed a plethora of notable individuals who supported and motivated me towards this success. Some of them I have mentioned and the rest remain in my mind for the rest of my life.

AYAN CHAUDHURI

DATE:

Title of the Thesis

Addressing Nano-Technology through Synthesis of Reversible Logic using Quantum-Dot Cellular Automata

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

Introduction

1 INTRODUCTION

Keeping pace with the dictum provided by Gordon Moore, CMOS has virtually reached limitations, both fundamental and technological. The last decade has witnessed CMOS nearing its saturation limits. The savior in future computer architectures lies in exploring alternative materials for substituting CMOS and eliminating the challenges faced by CMOS. Added to the technological growth was the thrust towards Quantum Computing. Rise of interest in Quantum Computing has popularized the concept of Reversible Logic. Much of the challenges faced by CMOS today can be eliminated by the use of Reversible Logic. In search of components that can relate the concept of reversibility, many propositions have been made. Two notable propositions have been the fundamental reversible gates, i.e. Toffoli Gate [**1**], and the Quantum Dot Cellular Automata cells [**2**] [**3**]. These two basic designs have been tried to implement computations possible at the nanostructure level with near zero heat dissipation. The concept of reversibility has reflected [**4**] that theoretically zero heat dissipation is possible if the computations are made reversible. Heat dissipation is the biggest issues faced by CMOS as more and more transistors are being packed within a limited area. The transition of bits from logic '1' to logic '0' and vice versa is termed as a switching cycle. The heat generated during this single switching cycle amounts to $KTln2$ Joules [**5**]. A single normal computation consumes huge amount of switching cycles. Hence the heat generated during this simple computation is considerably high and that too when generated within a smaller area, may damage the chip.

Quantum Dot Cellular Automata cells possesses near zero heat generation attributes as the information transfer takes place through Coulomb force interactions between the electrons. The use of Toffoli gates too dissipates near zero heat due to the fact that the switching between logics takes place through magnetic interactions. Hence, these two technologies address the heat generation issue of CMOS devices. The major advantage of Quantum Dot Cellular Automata cells lies in the fact that a large amount of these cells can be packed with a smaller chip. It is to be noted that although QCA provides logical reversibility, information propagation is unidirectional

This thesis concentrates provides research related to Quantum Dot Cellular Automata (QCA) cells, architecture designed using QCA cells, provides a taxonomy for Quantum Dot Cellular Automata research, proposal for a reversible gate and its QCA implementation followed by arithmetic architecture designed using the reversible gate proposal.

1.1 CONCEPT OF REVERSIBLE LOGIC

The concept of reversible logic states that the computation through a series of units should be bidirectional. In other words, if the output of a unit is provided, the input should be computable from the outputs. In classical electronics, only the inverter or the NOT gate is reversible in concept. That is, the input of the NOT gate can be predicted if the output is provided. But that too is not bidirectional as information flow thorough CMOS is unidirectional. Hence by concept, the NOT gate is reversible but by implementation the process is irreversible. A library of reversible gates exist which exhibit reversibility both conceptually and through process also.

Since, the input should be predicted from the output, hence that requires that the number of inputs be equal to the number of outputs. The following are the requirements for a circuit or a unit to be reversible.

- The number of inputs should be equal to the number of outputs.
- The output should not be branched. That is, maximum possible fan out is 1.
- There should be a one-to-one mapping between the inputs and the outputs.
- There should be no output left which has been not mapped to the input.
- There should be no feedback from the output to the input of a single fundamental reversible gate.

Since, information is not lost through the intermediate steps of computations, i.e. information is preserved throughout the computation process; there is opportunity for lossless information propagation. This lossless information propagation accounts for near zero heat dissipation. Moreover the materials that are used to fabricate the reversible gates itself do not dissipate heat as information propagation is either magnetic or by the use of Coulomb interaction. The fundamental reversible gate library consists of three gates as shown in Figure 1.1.

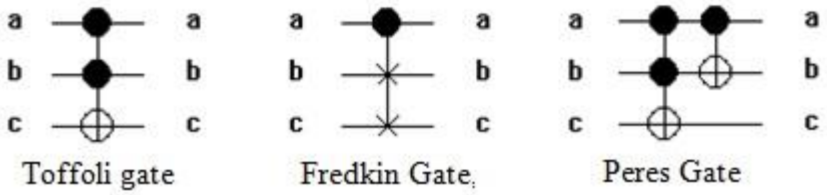


Figure 1.1. Fundamental reversible gates

Quantum Dot Cellular Automata cells partially exhibit the concept of reversibility as they can be designed to generate the input from the output with certain modifications in the information propagation method, i.e. the clocking sequence, as will be seen later in the thesis.

1.2 QUANTUM DOT CELLULAR AUTOMATA CELL

In contrast to classical electronics where information flow is triggered by physical flow of electrons, Quantum Dot Cellular Automata (QCA) cells aid in information flow by orientation of electrons, through re-orientation within the cells. Figure 1.2 shows a QCA cell. Figure 1.3 presents the schematic for a four electron QCA cell [**6**].

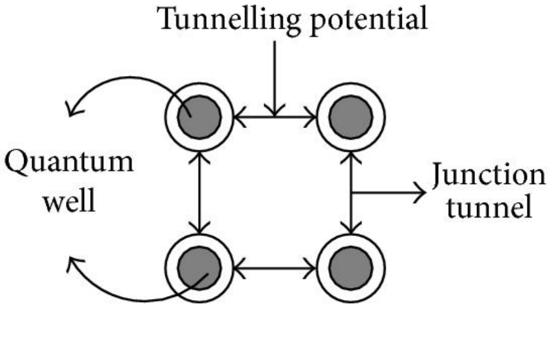


Figure 1.2. QCA Cell

The cell is an exactly square device having four corners called Quantum Dots. These Quantum Dots are islands or potential wells. The diameter of a QCA cell has been kept sufficiently small so that the charging energy is greater than $k_B T; where\ k_B = Boltzmann'sConstant\ and\ T = \ operating\ temperature.$

In these four potential wells (Quantum Wells), four electrons reside which can move under specific condition. The four potential wells are interconnected through passages known as Junction tunnels. Under specific conditions these junction tunnels are opened for the electron to pass through them. Two electrons residing in the four potential wells take the diagonal positions due to the Coulomb force that acts between them. These Coulombic repulsions make the electrons stay as far as they can from each other. They travel to their nearest blank potential well through the Junction tunnels. The electrons move to the dots through tunneling paths which are represented by solid bidirectional arrow connections between two islands in Figure 1.2.

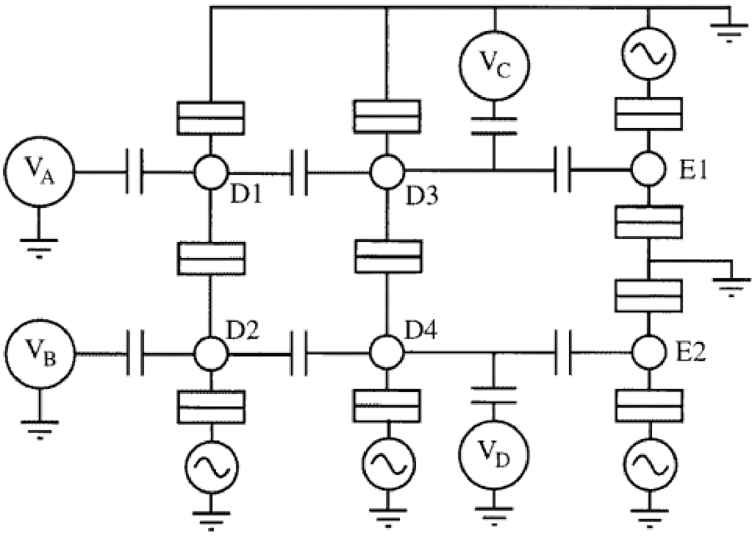


Figure 1.3. Schematic for a four electron QCA cell with two electrometers [**6**]

The junction tunnel area is very important as this area is considered for the charging capacitance which in turn determines the operating temperature of the QCA cell. Moreover the capacitances are varied thereby opening and closing of the tunnel gets place. It is by varying the junction potential that the electrons are allowed to flow through them and change the logic state of the QCA cell. Lent and

Tougaw in [2] have also proposed the replacement of electrons with metal dots and the tunnel junctions to be made up of aluminium oxide. This proposition follows from the fact that the crystalline lattice ions can effectively electrically neutralize the metal dots.

1.3 QCA CELLS AS LOGIC DEVICES

As stated earlier, Quantum Dot Cellular Automata Cells can be used to store logic values through the two electrons within a QCA cells with diagonal alignments. Figure 1.4 presents the schematic for representation of logic '0' and logic '1' using a standard QCA cell. The electron alignment in a QCA cell is governed by Coulomb forces as discussed earlier.

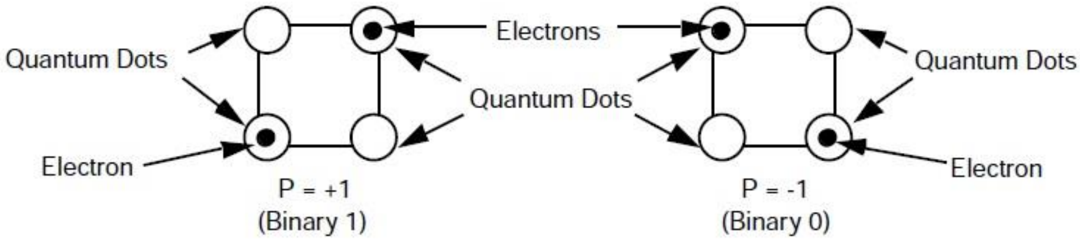


Figure 1.4. Electron orientation in a QCA cell denoting logic '0' and logic '1'

Existence of two electrons in four potential wells gives rise to two polarizations ($P = +1; P = -1$). Logic '1' is represented by $P = +1$ polarization and logic '0' is represented by $P = -1$.These two polarizations reflect information bit analogous to classical bit. When the electrons travel from one polarization to another polarization, information transfer takes place. Hence, unlike classical electronics, where information transfer is done by physical movement of electrons through a channel, in information transfer through QCA cells, electrons move within the QCA cell by changing the alignment. Their change in alignment is caused by Coulomb forces.

1.4 INFORMATION PROPAGATION THROUGH QCA CELLS

As stated earlier, the information propagation through QCA cells takes place by

Coulomb forces applied to one QCA cell due to another QCA cell. Figure 1.5 represents the cell-to-cell response while information propagation through a QCA cell.

In figure 1.5, it can be observed that when a single QCA cell is at logic 0 ($P = -1$), it affects the neighboring QCA cell to be at logic 0. With change in formation content of the QCA cell from logic 0 to logic 1 ($P = +1$), the neighboring cell also changes to logic 1 due to Coulombic interaction between electrons within the first QCA cell and the electrons within the next QCA cell. As the polarization between the electrons within one QCA cell changes from one to another, The electrons get re-aligned and then they come to the electrons of the neighbor QCA cell. Here, they force the electron in that cell to re-align accordingly to have maximum distance between them. Thereby the information change caused within one QCA cell gets propagated to the neighboring QCA cell without actual electron travel. The electron travel is constrained within a single QCA cell only.

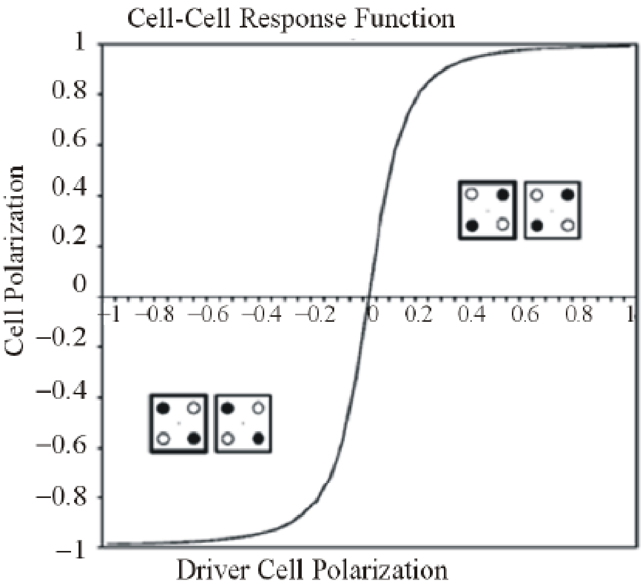


Figure 1.5. Change in junction capacitance and QCA cellular response [7]

The lower part of Figure 1.5 describes the two QCA cells having logic 0. The upper part of Figure 1.5 reflects the two electrons having logic 1, and hence the polarizations.

There are two types of QCA cells in practise. The regular QCA cell, which has been already discussed, and the other type is the symmetric type. Both the types are shown in Figure 1.6

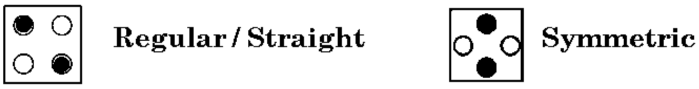


Figure 1.6. Regular and Symmetric QCA cells

Regular cells pass information through a single logic. That is, in regular cells, if the first cell has logic 0, then the adjacent cells also have logic 0. In case of symmetric cells, the information flow is aided by signal inversion. If a QCA cell has logic 0, the adjacent QCA cell has logic 1. These are reflected in Figure 1.7.

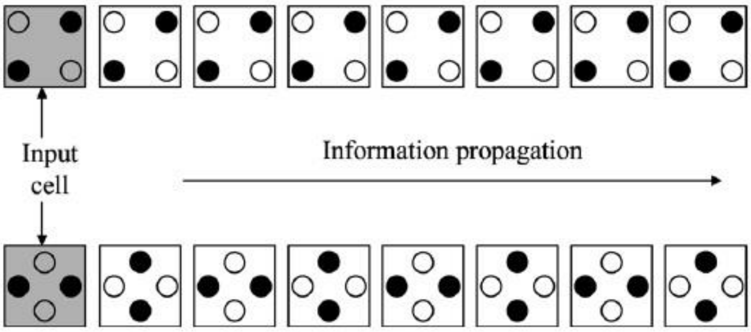


Figure 1.7. Information propagation through two types of QCA wires

Lower part of Figure 1.7 reflects the signal inversion concept during information transfer through a symmetric QCA wire. Continuous placement of QCA cells are known as QCA wires. Information propagation in QCA wires are also termed as Information transformation due to the fact that each QCA cell is getting transformed from one logic to another during information propagation.

1.5 MAJORITY VOTER

The fundamental gate in QCA is known as the Majority Voter gate. It is designed using five QCA cells and based on the logic that the highest cumulative Coulombic force prevails. The total Coulombic force is the cells are added and the resultant polarization is fed to the output. Figure 1.8 presents the schematic for the Majority

Voter gate.

By definition “*A reversible gate is termed as a Majority gate if it realizes a Majority Boolean Function (MBF). An MBF is a specification which has odd number of inputs and produces an output which has the majority votes among the inputs*”.

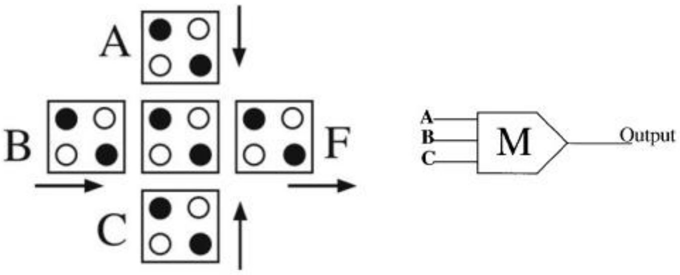
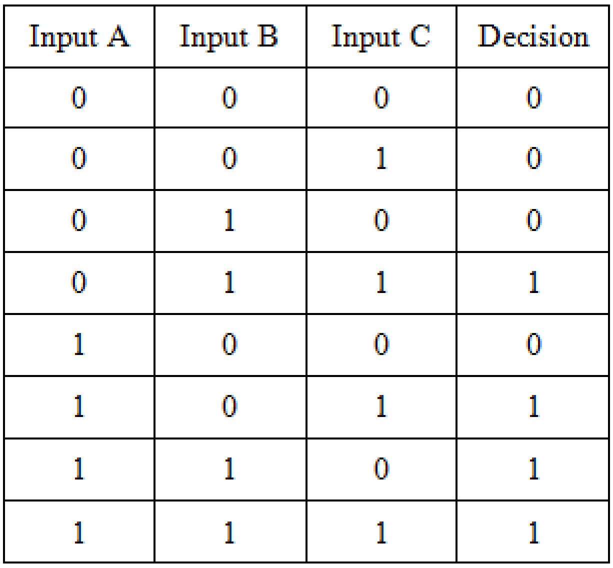


Figure 1.8. Schematic for Majority Voter

In Figure 1.8, A, B, and C are the three inputs. The central cell is called the decision cell. The decision is taken depending upon the total Coulombic force of the three inputs. Logically speaking, the decision cell takes the value of two or more of the input cells. That is, if two or more inputs are at logic 1, then the decision cell is at logic 1. Table 1.1 presents all the possible outcomes using the three inputs and the value of the decision cell.

TABLE 1.1. DECISION TABLE VALUES



Input A	Input B	Input C	Decision
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

From Table 1.1, the expression for the decision cell can be deduced using K-Map method. Equation 1 presents the equation from the decision function using the three inputs A, B, and C.

$$Decision = AB + BC + CA \ (1)$$

Equation (1) reflects the expression for the carry output for the expression of a full adder circuit. Hence, while designing a full adder, a single QCA cell is enough to design the Carry output for the full adder.

Using equation (1), a single majority voter can be used to design the gates for Boolean logic implementing the AND, OR, and NOT operation as shown in Figure 1.9, Figure 1.10, and Figure 1.11.

For generating the AND logic, one of the three inputs of a majority voter is fixed at logic 0. Hence the decision cell has logic 1 only if the other two inputs are at logic 1. Similarly for implementing the OR gate, one of the inputs are fixed at logic 1. Therefore, if any one of the rest two inputs are at logic 1, then the decision cell has logic 1.

The value of the decision cell is then passed to the output cell F, as shown in Figure 1.8.

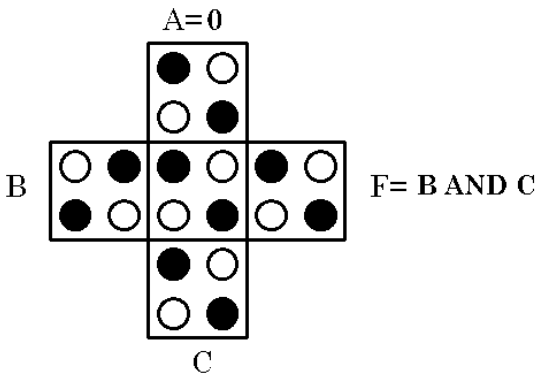


Figure 1.9. AND operation implemented using Majority Voter gate

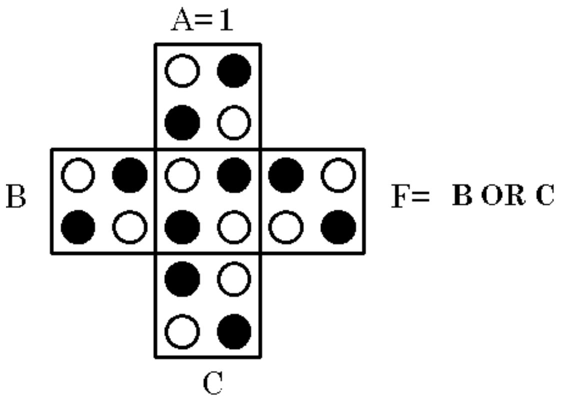


Figure 1.10. OR operation implemented using Majority Voter gate

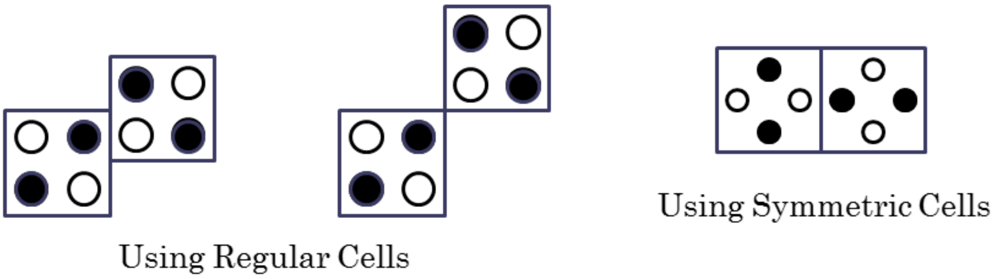


Figure 1.11. NOT operation implemented using Majority Voter gate – regular and symmetric cells

Figure 1.11 presents the designs for inverters using regular as well as symmetric cells. While working with QCA cells, it is important to note that the Coulombic effect of a certain QCA cell extends beyond the single neighbouring cell.

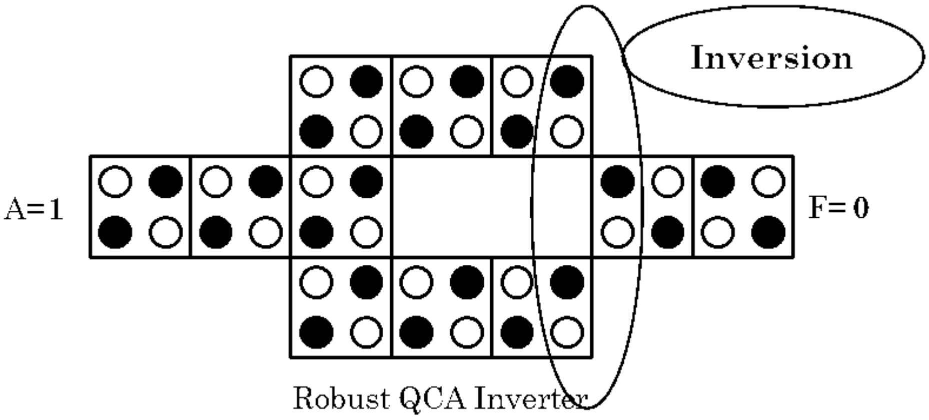


Figure 1.12. Robust QCA inverter

Hence, the effect for cell 1 in a wire can be felt at cell 3 or cell 4 as the case may be. Thus to be very sure of the inversion operation, a more robust QCA inverter is usually used as shown in Figure 1.12. The point of signal inversion is marked in Figure 1.12.

Having designed the QCA NOT gate, the QCA AND, and the QCA OR gates, designs for the universal gates of Boolean algebra are provided in Figure 1.13 and Figure 1.14 respectively, i.e. the QCA NAND and the QCA NOR gate.

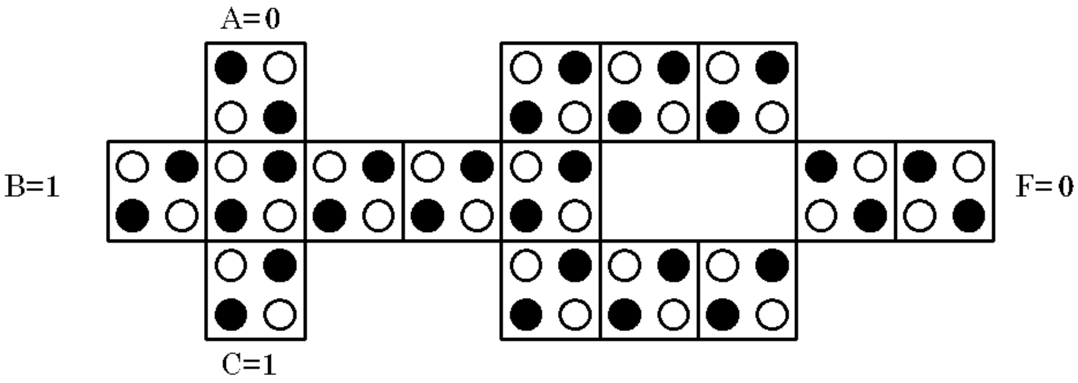


Figure 1.13. Robust QCA NAND Gate

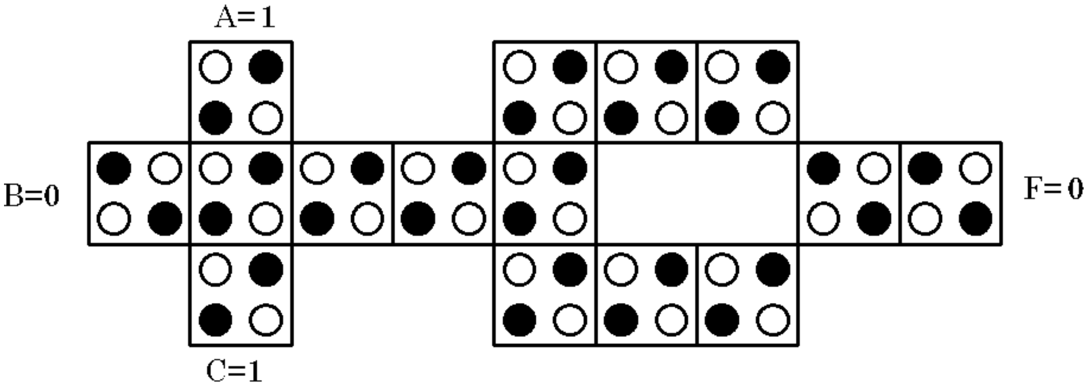


Figure 1.14. Robust QCA NOR Gate

1.6 QCA WIRES AND FAN-OUT

As discussed earlier, series of QCA cells are termed as QCA wires [**8**]. It is interesting to note the wire crossings in such a circuit. Although multilayer QCA wire architectures exist, but it is in a very nascent stage. Hence this thesis and the architecture design proposals concentrate on coplanar wire crossings. Figure 1.15 present the wire crossing most widely used in QCA circuits.

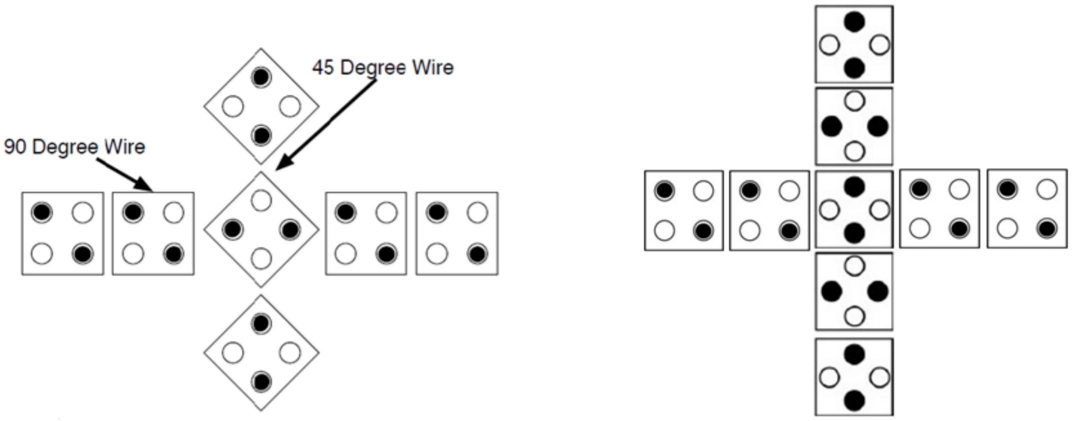


Figure 1.15. QCA wire crossing using regular as well as symmetric cells

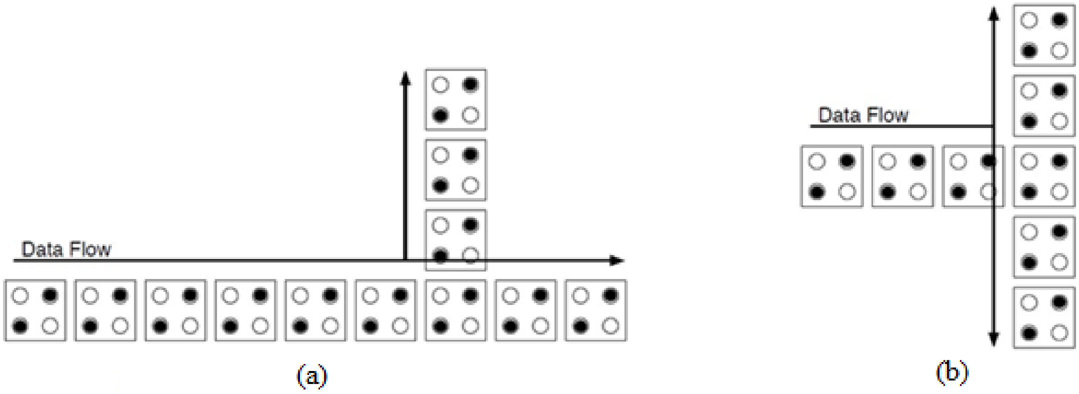


Figure 1.16. Fan-out from QCA wires (a) regular fan-out, (b) T - fan-out

Figure 1.16 presents the type of fan-outs possible using QCA wires. Figure 1.16a presents the regular fan-out whereas Figure 1.16b presents the T-Fan-out.

1.7 QCA CLOCKS

One of the underlying attributes for the popularity of QCA is the clocking of QCA circuits. Unlike traditional clocks used in classical circuits, The QCA clock is divided into 4 zones as shown in Figure 1.17. Figure 1.18 presents the stage of a QCA cell during different stages of the clock zone.

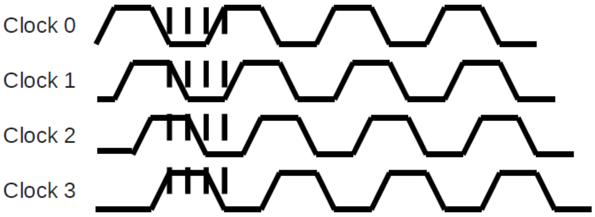


Figure 1.17. QCA clocking zones

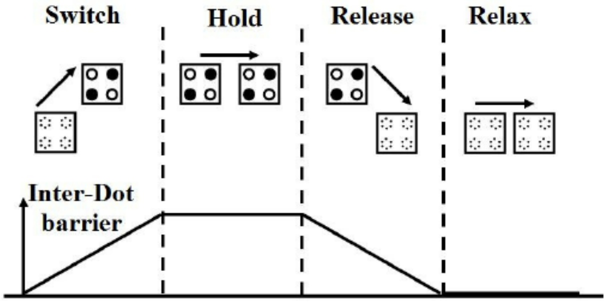


Figure 1.18. QCA Cell stages during different clocks

QCA clocks are divided into four clock zones – clock 0, clock 1, clock 2, and clock 3. These four stages are termed as Switch, Hold, release, and Relax for the QCA cell experiencing the clock zones. In the first clock zone, the QCA cell value switches to a new value. In the second clock zone, the value is retained by the QCA cell. The third clock zone helps the QCA cell to release the content while in the last clock zone, the QCA cell is in the floating cell with no garbage value residing in it. Hence, during clock zone 3 in a QCA cell, there is no valid value in that cell. Thus it can be concluded that a QCA cell helps in information propagation in the first three clock zones only.

Suppose there are four QCA cells placed adjacent to each other in a wire.

The first cell experiences a value change and subsequently the information propagates through the next cells. Figure 1.19 shows the potential held by each cell during information propagation along with the stages of the clock.

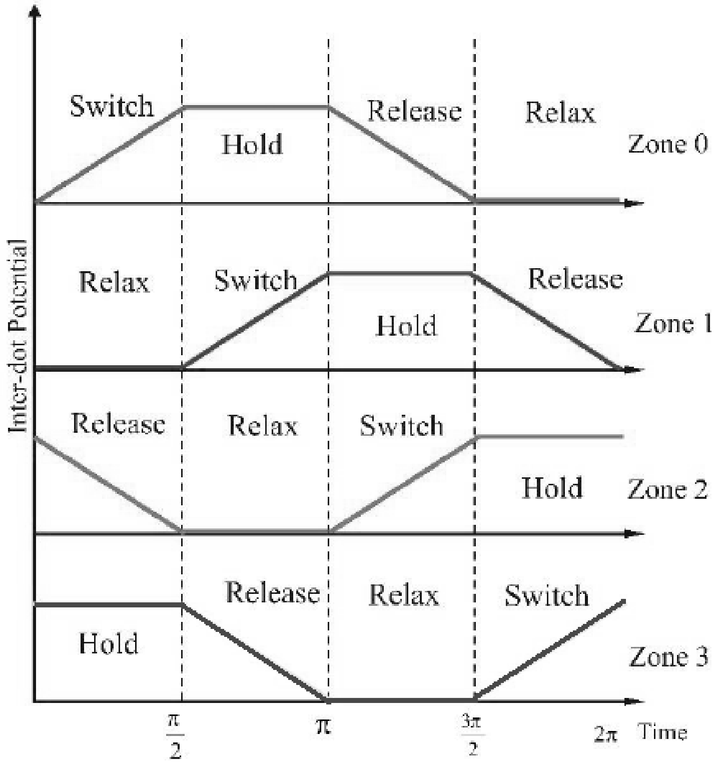


Figure 1.19. QCA cell status during different clock zones (for QCA wires)

Successfully propagation data takes place through a QCA wire when the clock zones in the wire is connected to successive clock signals in the direction of the desired data propagation. During high clock pulse, the tunnel junctions are opened allowing electrons to flow through them to specific dots in order to minimize Coulombic effect, thereby aligning itself in path of data transfer. This allows the QCA circuit designer to route data in a controlled manner through the circuit. Cell coverage of a single clock zone is design dependent. For example, large clock zones can be employed for QCA wires with no surrounding neighbors as the Coulombic noise is negligible. Noisy environments may demand clock zones to control as less as two cells. Figure 1.20 presents the clock shifts by the four clocks of QCA.

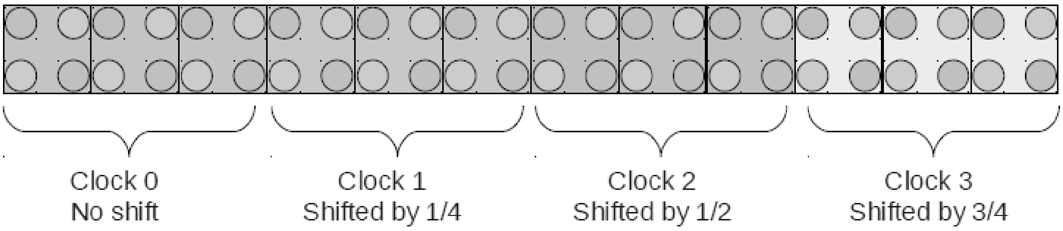


Figure 1.20. QCA clock shifts

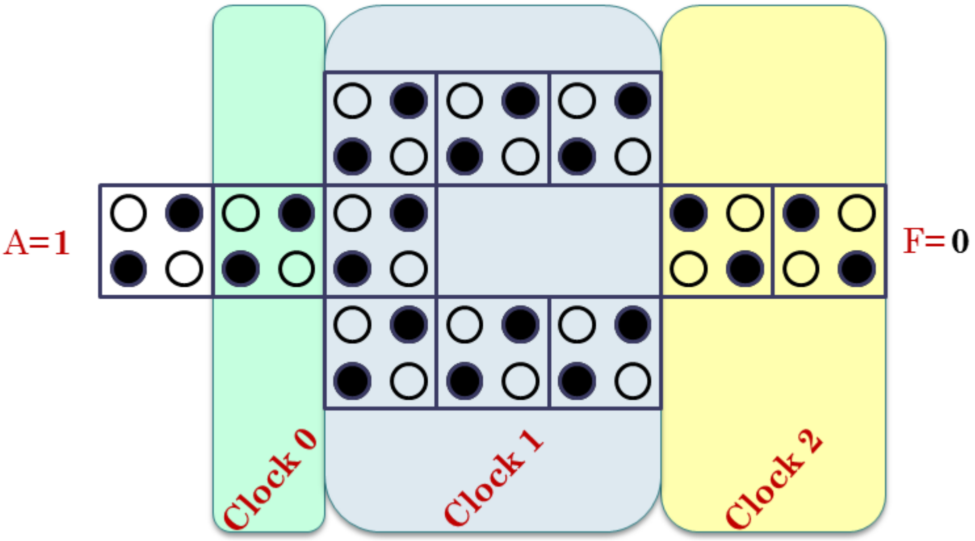


Figure 1.21. QCA clocks – Inverter clocking

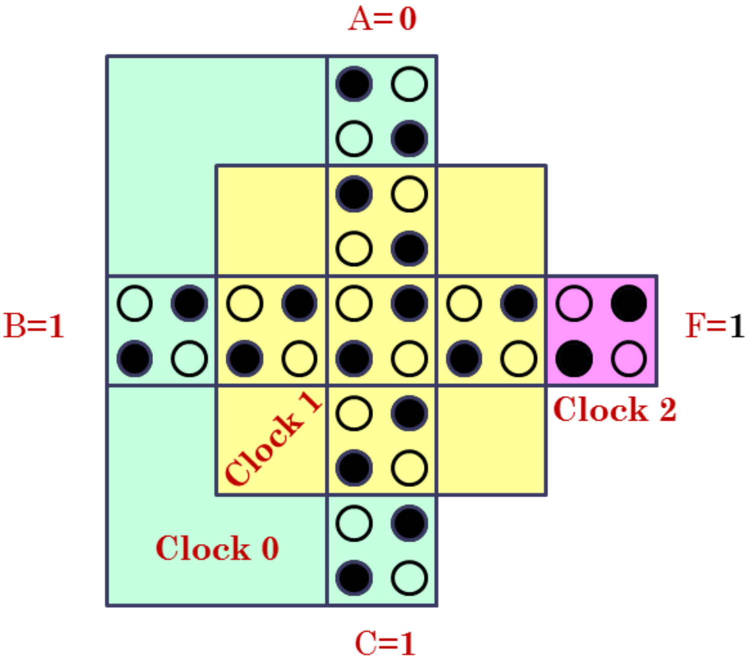


Figure 1.22. QCA clocks – Majority voter clocking

Figure 1.21 presents the QCA clock zones for a QCA inverter. Figure 1.22 presents the QCA clock zones for QCA Majority Voter. It is interesting to note that in Figure 1.22, the inputs are fed in clock 0, the decision is taken in clock 1, and the result is propagated in clock 2. However, the information propagation from decision cell to output ‘F’ can be done in clock 1 also, as has been practically observed using [**9**].

1.8 QCA APPLICATIONS AND RESEARCH DOMAINS

Quantum Dot Cellular Automata has found various applications with most of them concentrating on arithmetic and logical architecture deigns.

Adder designs have been provided in [**10**] [**11**] [**12**] [**13**] [**14**] [**15**] to name a few. Synthesis based proposals have been provided in [**16**] [**17**] [**18**] [**19**]. QCA memory units have been proposed in [**20**] [**21**] [**22**] [**23**] [**24**] to cite a few. There are other numerous dimensions in QCA research which will be discussed in Chapter 3 of this thesis.

The rest of the thesis is organized as follows.

Chapter 2 describes the QCA implementation for the reversible gates, namely RI [**25**], R2 [**25**], TSG [**26**], HNG [**27**], SCG [**28**], RPS [**29**], HNFG [**30**] [**31**], MKG [**30**], IG [**32**], FAG [**33**], ALG [**34**], MTSG [**35**], DFG [**33**], DKG [**36**], MRG [**36**], BG-GB Gates [**37**], s2c2 [**38**], TCG [**39**] [**40**] and BVMF [**41**] gates..

Chapter 3 provides a taxonomy for research on Quantum Dot Cellular Automata. This chapter has been motivated through the literature study. The study has been communicated in [**42**].

Chapter 4 presents a Hamming Code Generator/Checker circuit using QCA. It is the logical dimension for research on the topic. The study has been published in [**43**].

Chapter 5 provides the comprehensive analysis of CNF and XOR-AND representations. The study has been published in [**44**].

Chapter 6 provides the reversible gate proposal with its QCA implementation followed by an architecture implementation. This chapter provides an arithmetic dimension of the study and proposes circuits based on QCA. The study has been published in [**39**] [**40**].

Chapter 7 concludes the thesis with research extension prospects.

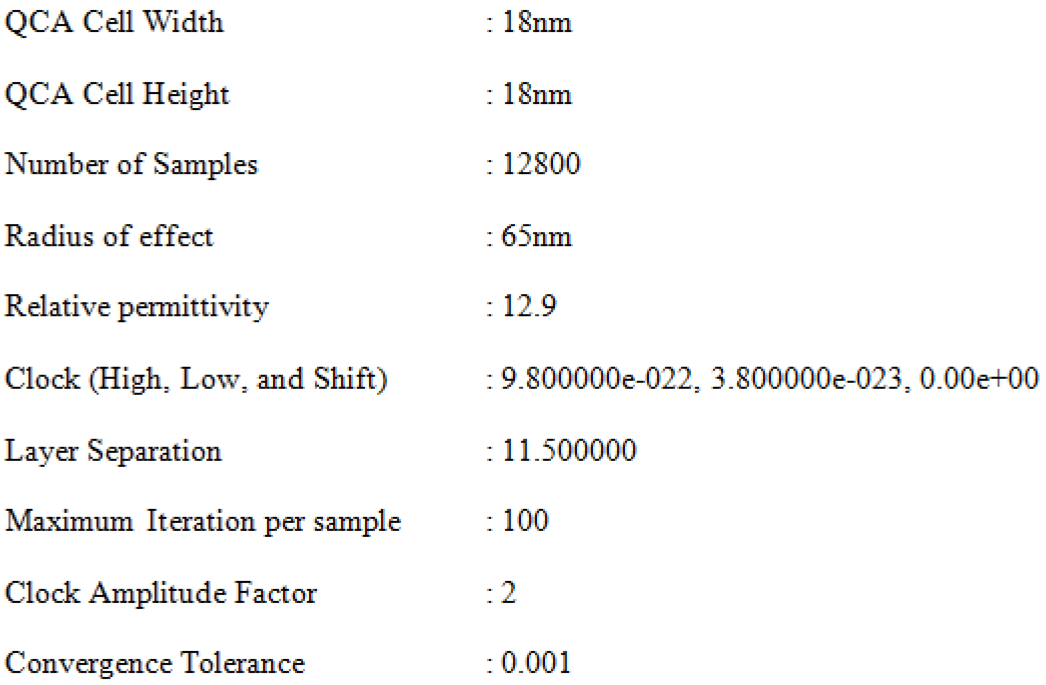
CHAPTER 2

QCA Implementation of Reversible Gates

2 QCA IMPLEMENTATION OF REVERSIBLE GATES

Quantum Dot Cellular Automata (QCA) forms one of the dimensions of Quantum Computing which operates partly on the concept of reversible logic. Reversing the clocking phases of the QCA circuits theoretically reverses the logic as in the case of reversible logic. One of the important properties of quantum computing is the property of reversibility. In this regard, both the domain of QCA and reversible logic witnessed massive research over the past one decade. Although conceived in 1993, QCA has gained interest in the past decade as will be shown in the next chapter. Since, reversible logic has been implemented using fundamental reversible logic gates, as discussed in Chapter 1; this chapter provides the details regarding the QCA implementations of the proposed reversible gates till now [**45**] (mainly the reversible gates having four inputs and four outputs).

This chapter provides the analysis of all the reversible gates with respect to the QCA cell properties. The designs have been simulated using QCA Designer [**9**]. The cell area is calculated by multiplying the QCA Cell Height with the QCA Cell Width. The simulation engine of QCA Designer has been set up using the parameters stated below. The initial sampling rate which has been used throughout the execution is 12800.



QCA Cell Width	: 18nm
QCA Cell Height	: 18nm
Number of Samples	: 12800
Radius of effect	: 65nm
Relative permittivity	: 12.9
Clock (High, Low, and Shift)	: 9.800000e-022, 3.800000e-023, 0.00e+00
Layer Separation	: 11.500000
Maximum Iteration per sample	: 100
Clock Amplitude Factor	: 2
Convergence Tolerance	: 0.001

All the proposed reversible gates have been proposed keeping in mind the application that they are intended to serve. For example full adder, full subtractor, comparator etc. Hence, this chapter details the notable reversible gates only and presents the QCA implementations for the same. QCA majority voter has been used extensively to design the circuits.

The notable reversible gates are RI [**25**], R2 [**25**], TSG [**26**], HNG [**27**], SCG [**28**], RPS [**29**], HNFG [**30**] [**31**], MKG [**30**], IG [**32**], FAG [**33**], ALG [**34**], MTSG [**35**], DFG [**33**], DKG [**36**], MRG [**36**], BG-GB Gates [**37**], s2c2 [**38**], TCG [**39**] [**40**] and BVMF [**41**] gates.

The next subsections provide the QCA realization of each of the reversible gates.

2.1 QCA IMPLEMENTATION OF R1 GATE

QCA design for the reversible R1 gate is presented in Figure 2.1.

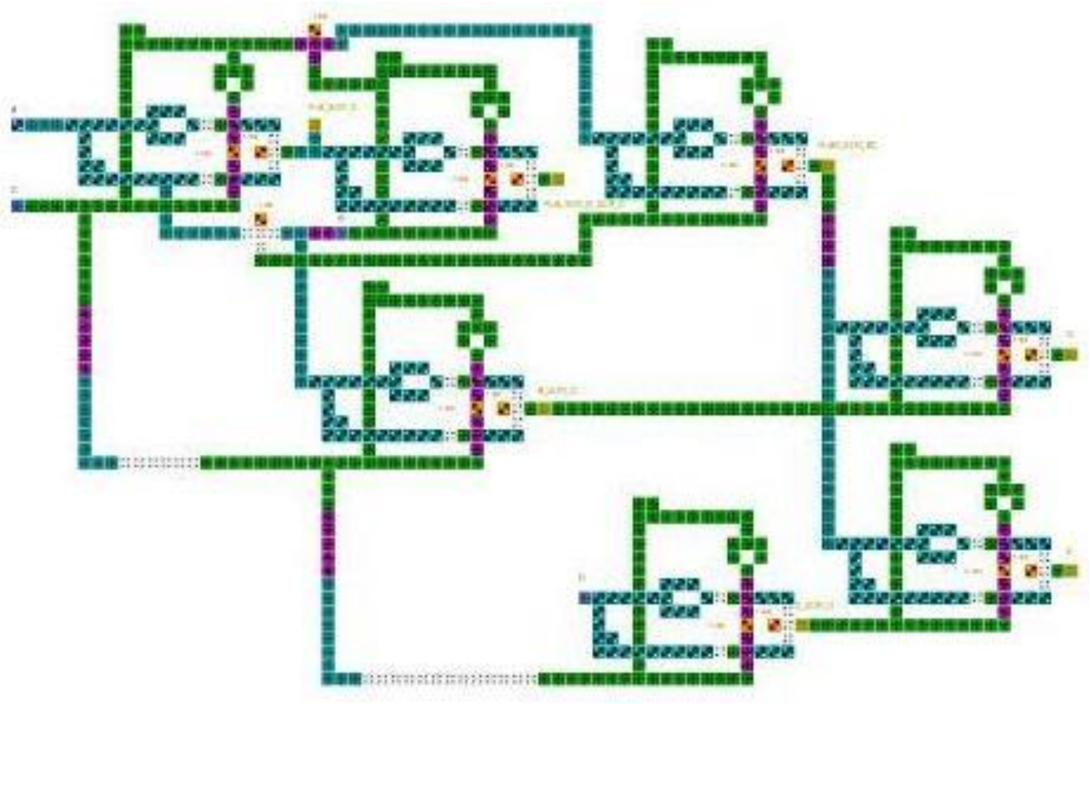


Figure 2.1. QCA Design of reversible R1 Gate

2.2 QCA IMPLEMENTATION OF R2 GATE

QCA design for the reversible R2 gate is presented in Figure 2.2.

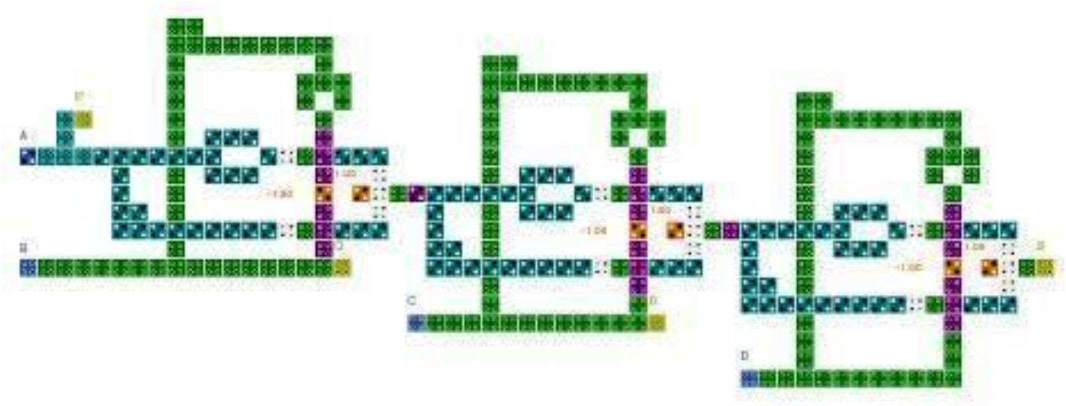


Figure 2.2. QCA Design of reversible R2 Gate

2.3 QCA IMPLEMENTATION OF DKG GATE

QCA design for the reversible DKG gate is presented in Figure 2.3.

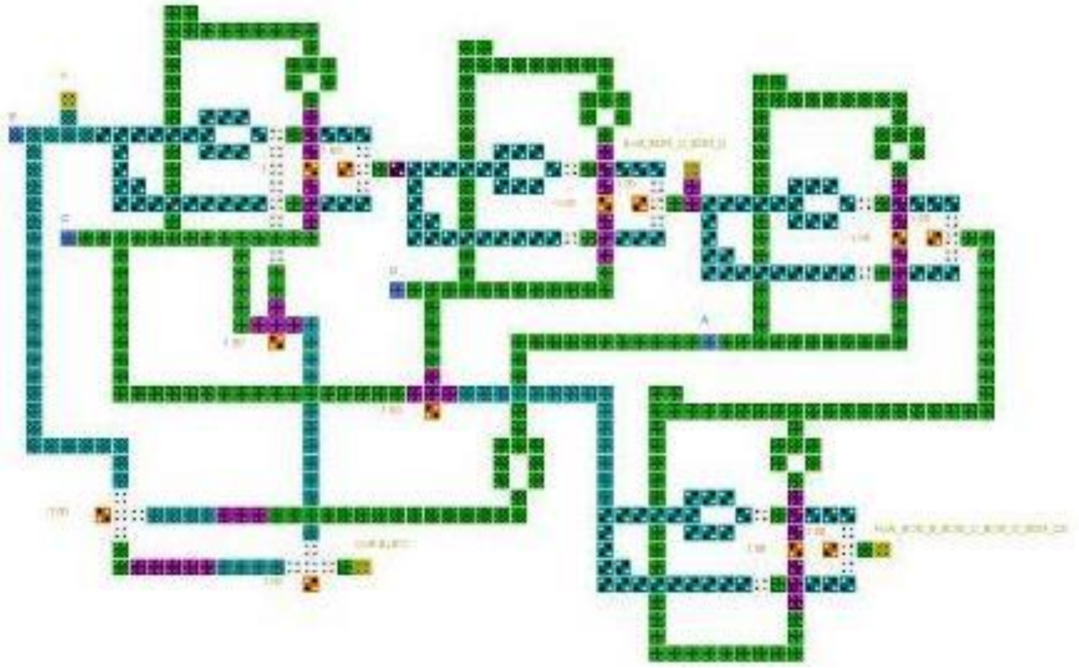


Figure 2.3. QCA Design of reversible DKG Gate

2.4 QCA IMPLEMENTATION OF DFG GATE

QCA design for the reversible DFG gate is presented in Figure 2.4.

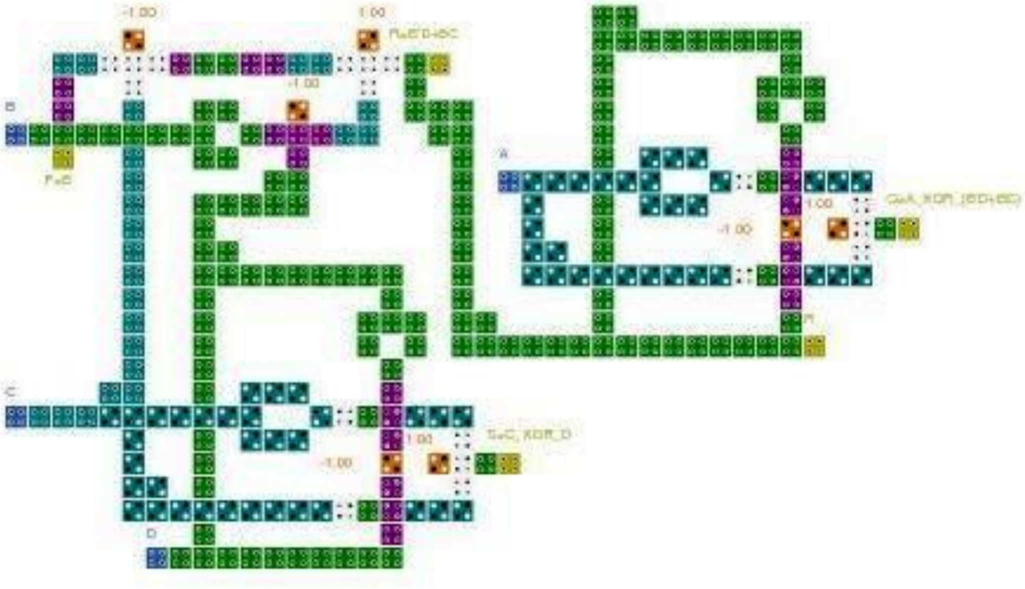


Figure 2.4. QCA Design of reversible DFG Gate

2.5 QCA IMPLEMENTATION OF TSG GATE

QCA design for the reversible TSG gate is presented in Figure 2.5.

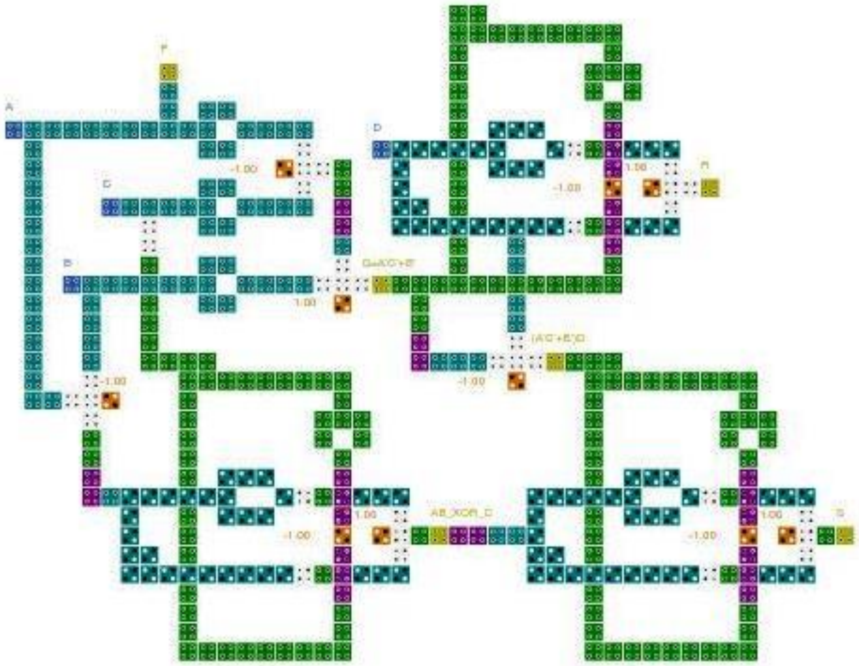


Figure 2.5. QCA Design of reversible TSG Gate

2.6 QCA IMPLEMENTATION OF FAG GATE

QCA design for the reversible FAG gate is presented in Figure 2.6.

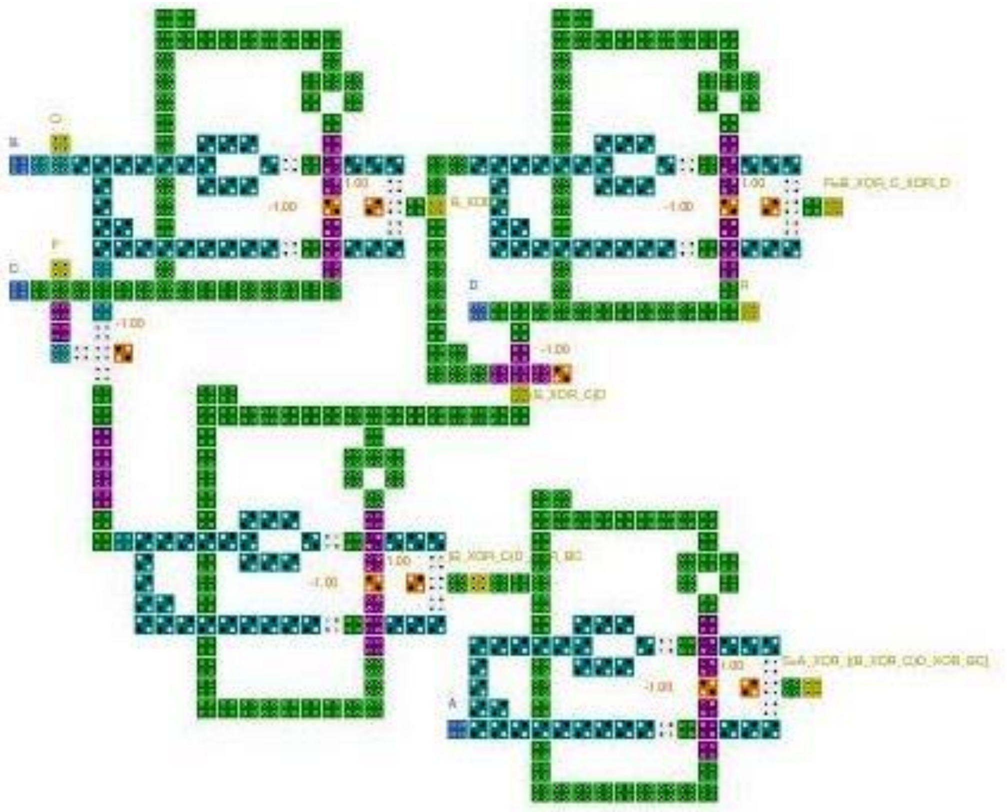


Figure 2.6. QCA Design of reversible FAG Gate

2.7 QCA IMPLEMENTATION OF HNFG GATE

QCA design for the reversible HNFG gate is presented in Figure 2.7.

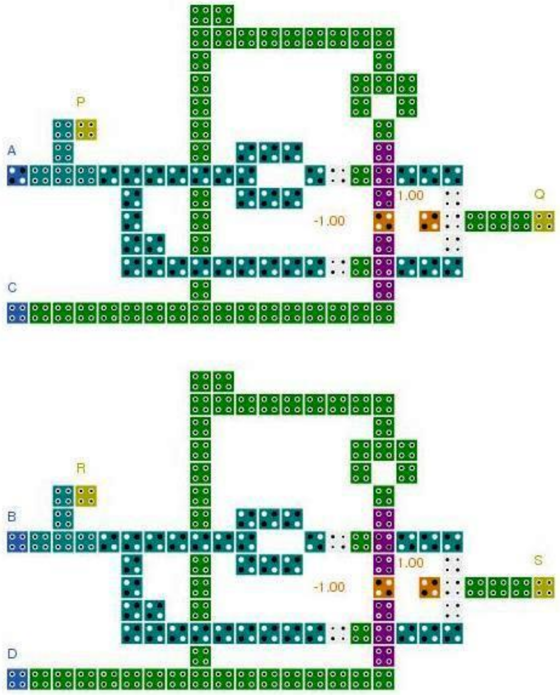


Figure 2.7. QCA Design of reversible HNFG Gate

2.8 QCA IMPLEMENTATION OF HNG GATE

QCA design for the reversible HNG gate is presented in Figure 2.8.

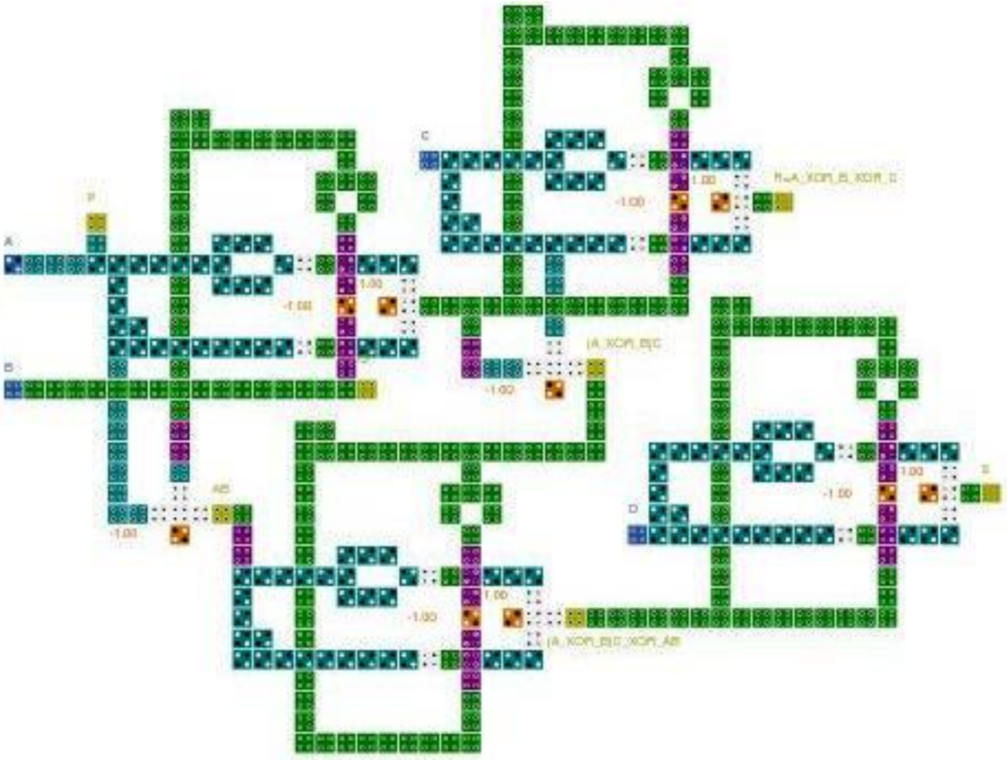


Figure 2.8. QCA Design of reversible HNG Gate

2.9 QCA IMPLEMENTATION OF RPS GATE

QCA design for the reversible RPS gate is presented in Figure 2.9.

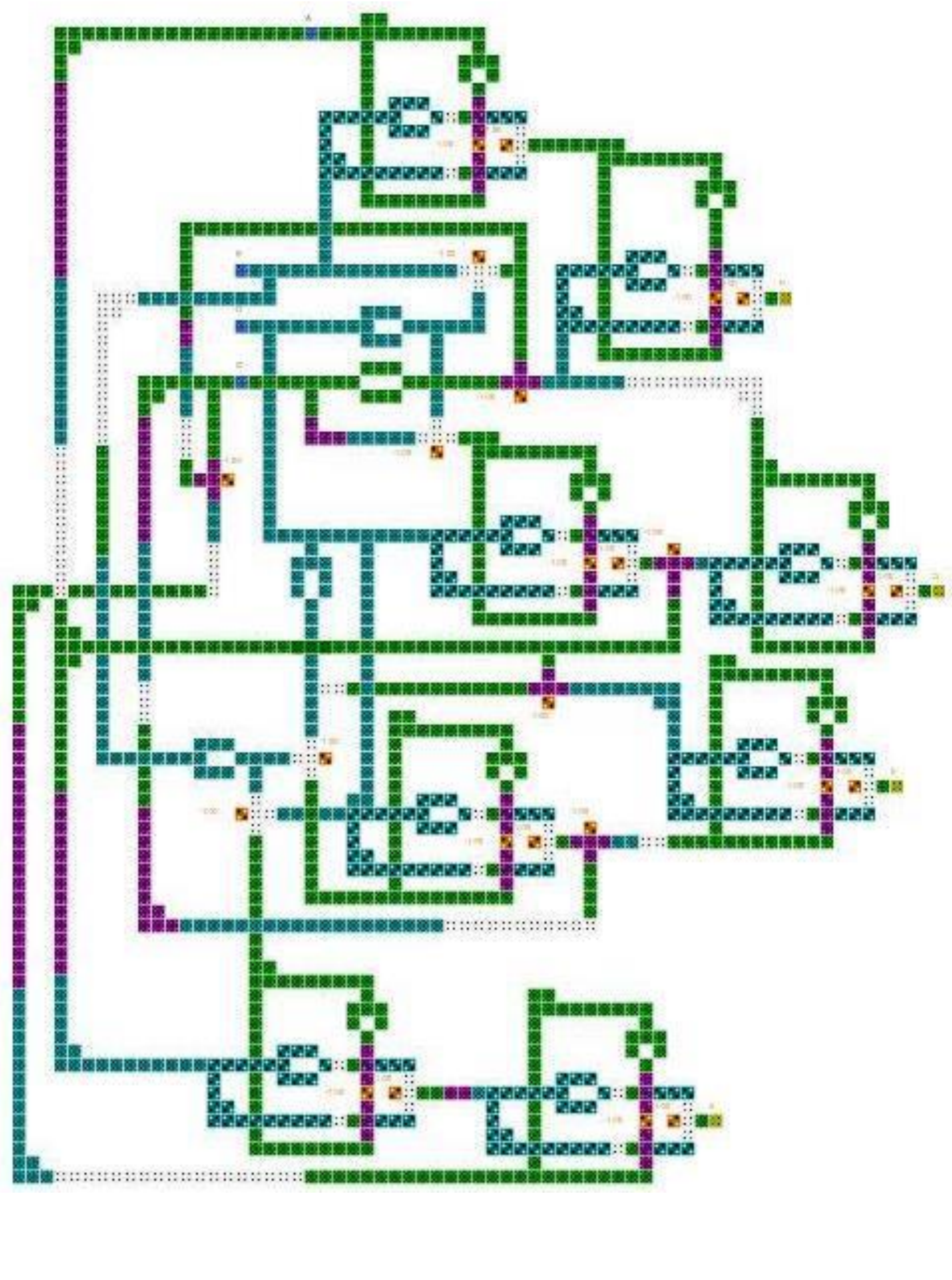


Figure 2.9. QCA Design of reversible RPS Gate

2.10 QCA IMPLEMENTATION OF SCG GATE

QCA design for the reversible SCG gate is presented in Figure 2.10.

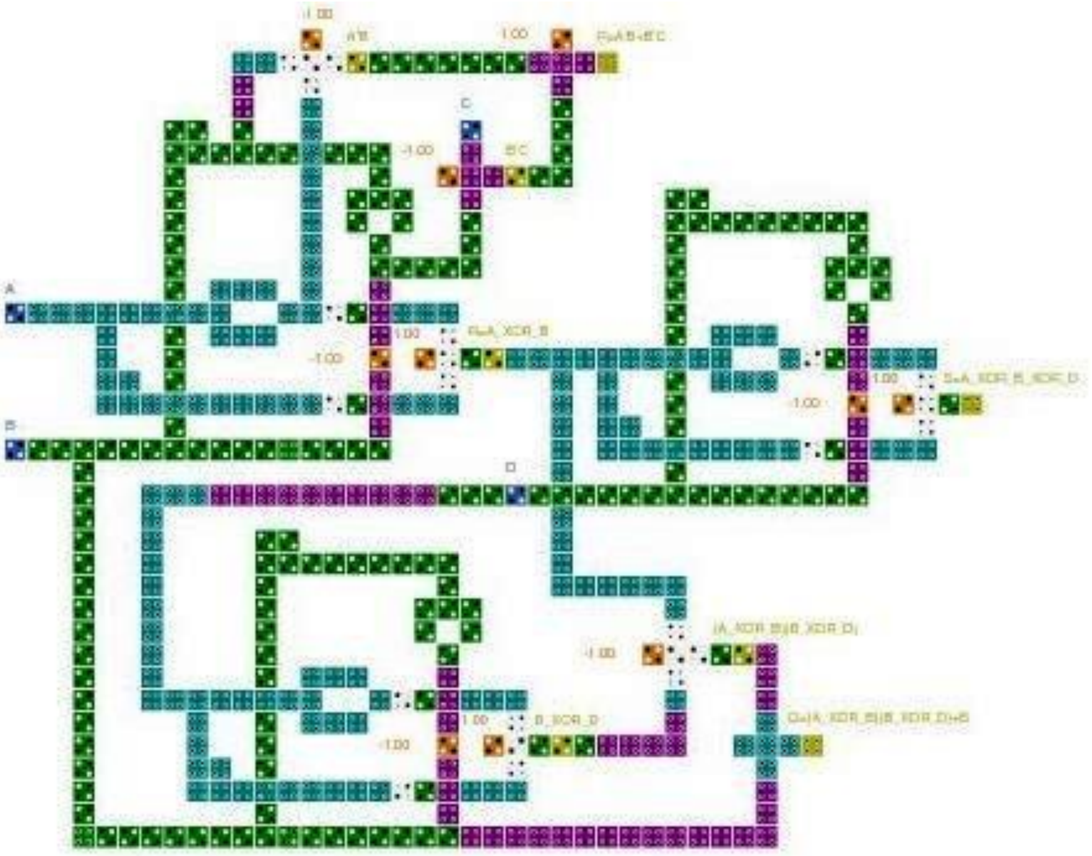


Figure 2.10. QCA Design of reversible SCG Gate

2.11 QCA IMPLEMENTATION OF MTSG GATE

QCA design for the reversible MTSG gate is presented in Figure 2.11.

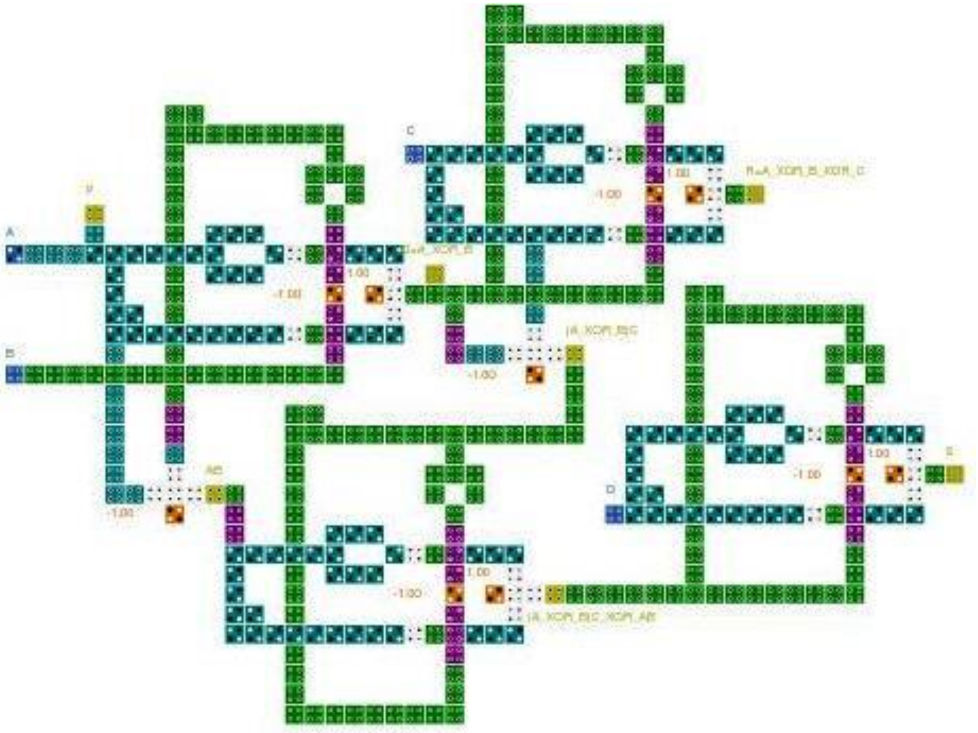


Figure 2.11. QCA Design of reversible MTSG Gate

2.12 QCA IMPLEMENTATION OF MRG GATE

QCA design for the reversible MRG gate is presented in Figure 2.12.

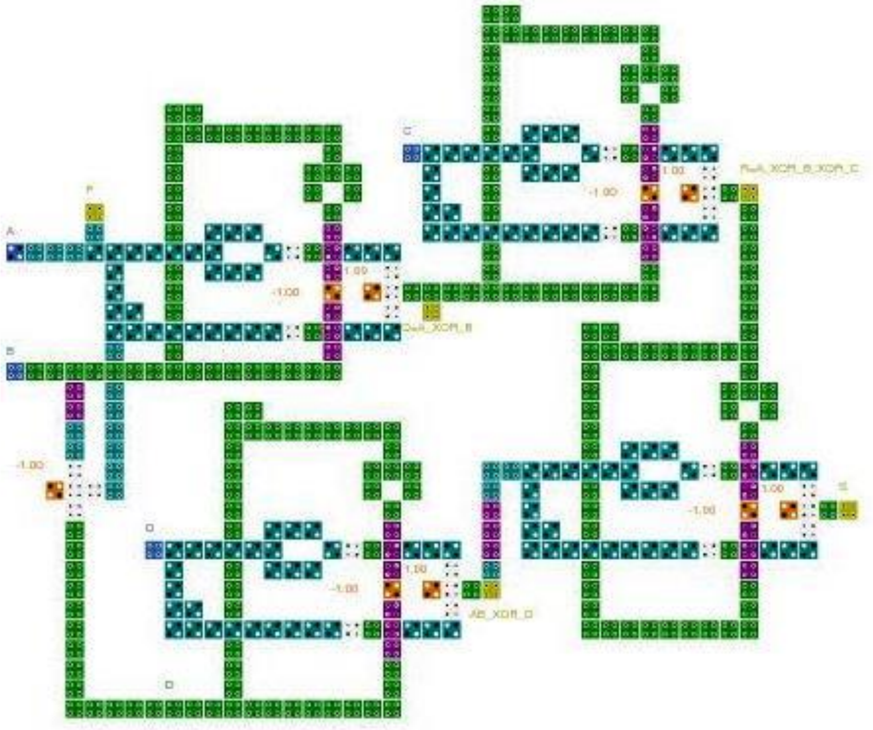


Figure 2.12. QCA Design of reversible MRG Gate

2.13 QCA IMPLEMENTATION OF MKG GATE

QCA design for the reversible MKG gate is presented in Figure 2.13.

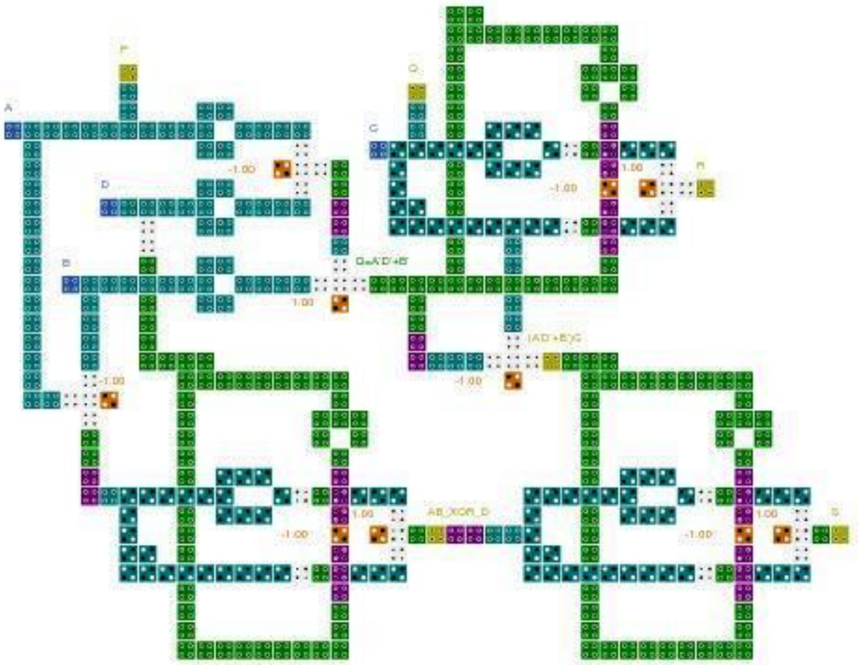


Figure 2.13. QCA Design of reversible MKG Gate

2.14 QCA IMPLEMENTATION OF IG GATE

QCA design for the reversible IG gate is presented in Figure 2.14.

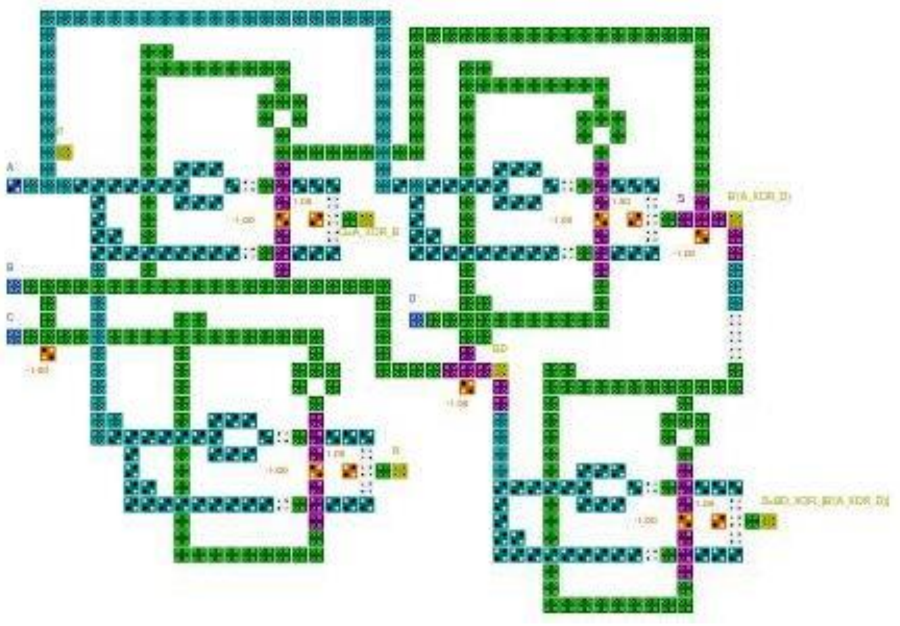


Figure 2.14. QCA Design of reversible IG Gate

2.15 QCA IMPLEMENTATION OF BVMF GATE

QCA design for the reversible BVMF gate is presented in Figure 2.15.

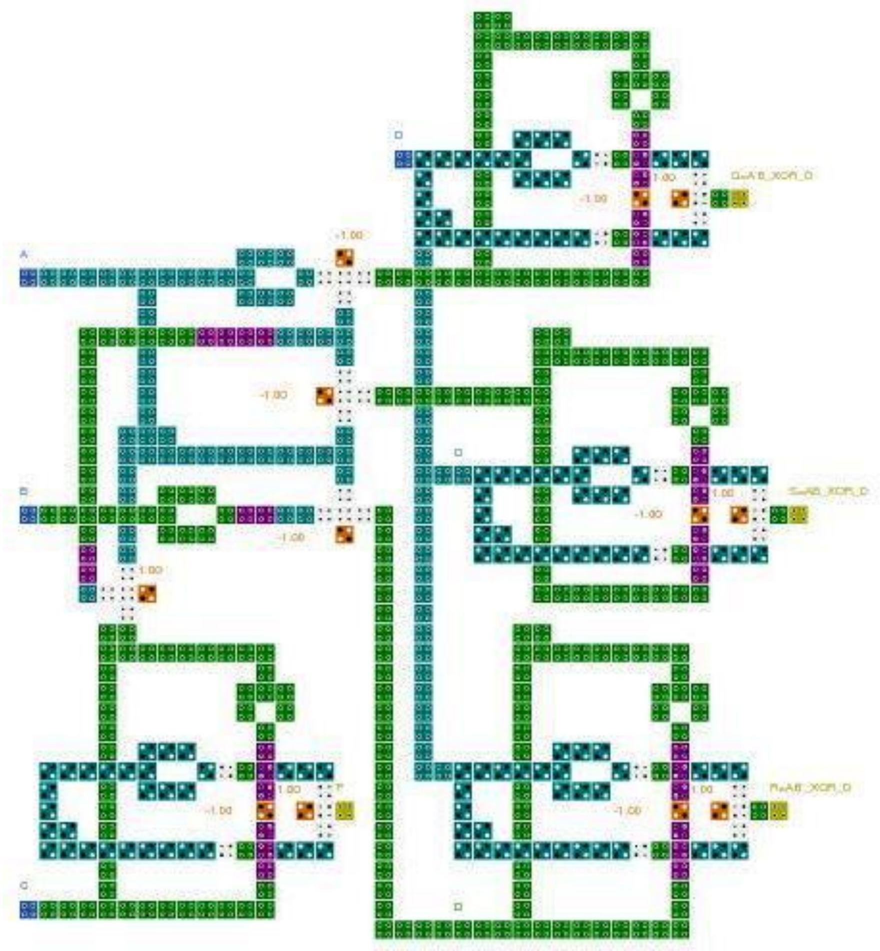


Figure 2.15. QCA Design of reversible BVMF Gate

2.16 QCA IMPLEMENTATION OF ALG GATE

QCA design for the reversible ALG gate is presented in Figure 2.16.

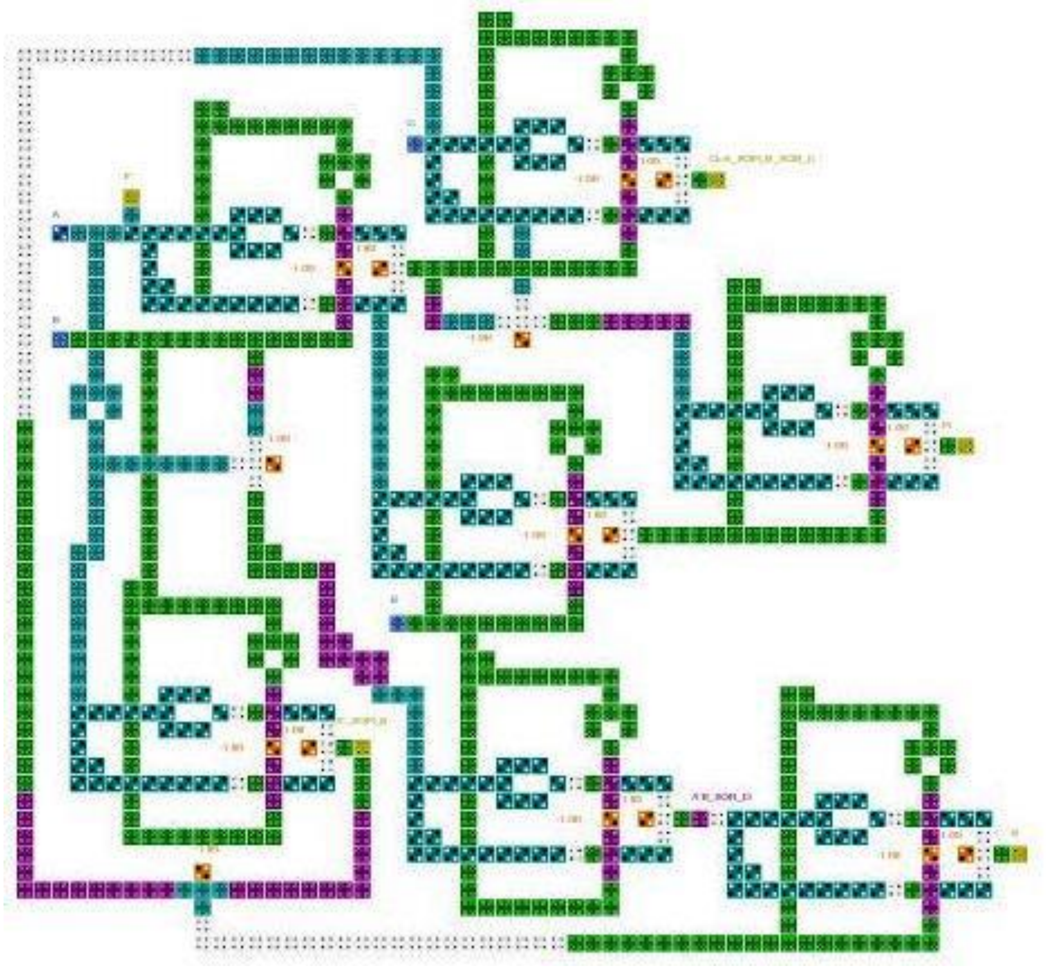
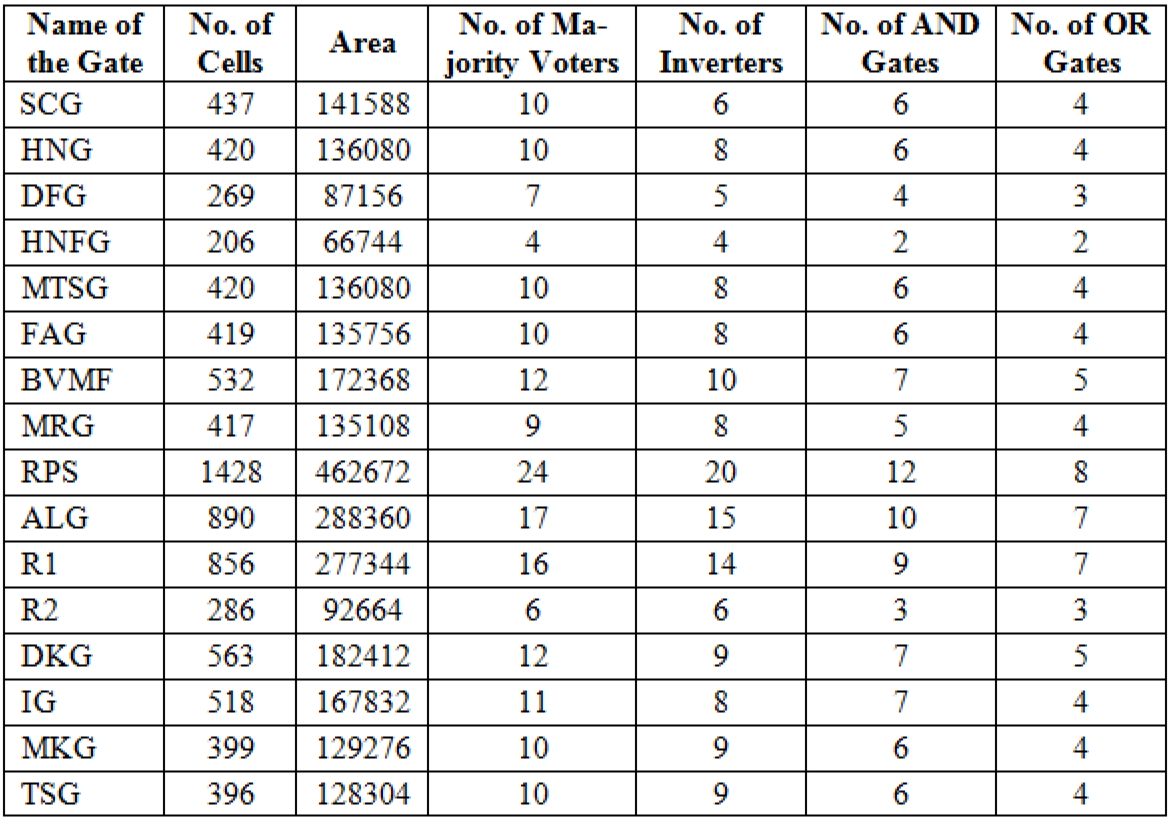


Figure 2.16. QCA Design of reversible ALG Gate

The comparative analysis for the QCA designs of the reversible gates (four variables) is provided in Table 2.1.

The details have been provided not only in terms of number of QCA cells and the area of the designs, but also in terms of number of Majority Voters, number of AND gates, number of OR gates, and the number of inverters required. This comparison can be used to compare designs with respect to the classical Boolean properties such as the number of fundamental Boolean gates required. Fundamental Boolean gates comprise of AND, OR, and NOT gates.

TABLE 2.1. COMPARATIVE ANALYSIS FOR THE QCA DESIGNS



Name of the Gate	No. of Cells	Area	No. of Majority Voters	No. of Inverters	No. of AND Gates	No. of OR Gates
SCG	437	141588	10	6	6	4
HNG	420	136080	10	8	6	4
DFG	269	87156	7	5	4	3
HNFG	206	66744	4	4	2	2
MTSG	420	136080	10	8	6	4
FAG	419	135756	10	8	6	4
BVMF	532	172368	12	10	7	5
MRG	417	135108	9	8	5	4
RPS	1428	462672	24	20	12	8
ALG	890	288360	17	15	10	7
R1	856	277344	16	14	9	7
R2	286	92664	6	6	3	3
DKG	563	182412	12	9	7	5
IG	518	167832	11	8	7	4
MKG	399	129276	10	9	6	4
TSG	396	128304	10	9	6	4

CHAPTER 3

Taxonomy for research on QCA

Ayan Chaudhuri, Mahamuda Sultana, Diganta Sengupta, Debashis De, Chitrita Chaudhuri, "An empirical study on published literature in QCA – A Taxonomy Proposal", IET Computers & Digital Techniques (SCI Indexed- Impact Factor 0.857) (Communicated).

3 TAXONOMY FOR RESEARCH ON QCA

While conducting literature survey it has been observed that a massive collection of research articles need to be consulted before shortlisting the articles of interest. Only a very small amount of relevant articles exist in tune to one's choice of domain. But the researcher has to go through a very large collection of data. This chapter proposes a taxonomy on research on QCA. The complete domain of research has been divided into several important nodes according to logic and then the references have been clustered together bearing relevance to a certain node. This helps the researcher in saving time otherwise used in literature study of the whole mass of published articles. Also researchers searching for a particular topic related to a certain domain will have ease in getting hold of the references within a short time.

With increasing volume of research output and expanding journal count, identification of relevant study for a certain domain of interest is becoming extremely challenging. Shortlisting of the most pertinent article in tune with one's interest forms the nontrivial task in literature study; thereby demanding extensive search. Facilitating such online article search, research journals demand authors to provide a certain amount of Author Keywords (AK), which should a. represent the manuscript content, b. be very precise to the study domain, and c. contain enough clarity to map the manuscript to specific editors as well as reviewers. In essence, AKs can be presumed to be brief pointers to the content of the manuscript [**46**]. The purpose of AKs is to make the articles more discoverable. Abstract and AK define the field, subfield, topic, and research issue; hence semantic analysis of the abstract provide the theme of the article whereas AK serve as batons for discoverability of the article. The work in this chapter concentrates in clustering references connected with certain AKs so that a researcher searching for references for a certain domain do not have to explore the whole mass of publications and can only concentrate on the cluster. For example, someone interested in 'clocking' of Quantum Dot Cellular Automata (QCA) cells can obtain the references clustered for 'clocking' rather than exploring the massive set of literature for QCA. The taxonomy (related cluster) proposal in this chapter also provides the relationship between AKs in a Parent-Child [P: C] manner. Thus, the researcher also gets hold of the pre-requisite cluster of references required to grasp/explore the knowledge of 'clocking' as 'clocking' forms the child of a certain Parent AK. The taxonomy also states the growth of research in QCA. It also provides the percentage of research related to AKs, i.e. research percentage of 'clocking' within the domain of QCA.

Generation of similarity between two AKs have been done using the Cosine Similarity function (CSF) as presented in Equation (1).

$$Cosine\ Similarity = \frac{n_{x,y}}{\sqrt{n_x}\sqrt{n_y}}\ (1)$$

Where $n_{x,y}$ = Number of articles containing both AK 'x' and AK 'y',

n_x = Number of articles containing only AK 'x', and

n_y = Number of articles containing only AK 'y'.

Similarity measures can be obtained using two functions, viz. CSF and Normalized Google Distance (NGD) [**47**]. Both CSF and NGD are undirected graphs. NGD primarily provides a similarity measure between two texts using the hit count in the Google search engine. CSF provides offline data processing in contrast to online data processing by NGD. This paper uses offline processing of data; hence the use of CSF by the authors. The undirected graph generated by CSF can be further used by Dijsktra-Jarnik-Prim's (DJP) Algorithm, Kruskal's Algorithm, Edmond's Algorithm, Heymann Algorithm and, Genetic algorithm [**48**] for further taxonomy analyses.

The taxonomy is basically the cluster of references according to AKs combined with the [P: C] relationship. These two properties transform the taxonomy into a tree structure with the top-most node as the root node. Taxonomy for classification and hierarchical dissemination of knowledge can be dated back to the 1970s [**49**]. The journal 'Nature' describes taxonomy as the – "Classification and description of living organisms. It includes the naming and defining of species and the collation of data about their biology and biogeography". This paper is an attempt to classify the different dimensions of research on QCA based on AKs; the target audience being the amateur and expert researchers in the domain.

3.1 BACKGROUND FOR TAXONOMY

Quantum Dot Cellular Automata has been a topic of extensive research over the last two decades [**50**] [**51**] owing to its alliance to the concept of logical reversibility [**52**]. Research in QCA propelled because information change in QCA cells lack heat generation otherwise bottleneck for CMOS devices. QCA also provides clocking speeds in THz range rather than GHz range for CMOS. Hence, QCA emerged as a viable alternative for CMOS in designing Boolean specifications at the architecture level. Several digital architectures have been conceived using QCA, i.e. Adder [**53**] [**54**] [**55**], Binary Subtractor [**56**], and applications in steganography [**57**]. Since, QCA forms an inclusive branch of emerging technologies; hence the authors have concentrated on generating the taxonomy for QCA.

As already discussed, although taxonomy proposals existed late 1970s, mostly were referred to dissemination of knowledge in biology, zoology and allied domains. Taxonomies for technical domains based on CSF can be observed in the taxonomy for Decimal Multiplier research [**58**], Reversible Logic [**59**], IoT based Home Automation [**60**] and Ambient Computing [**61**]. Authors in [**58**] have proposed a six-level taxonomy for research on Decimal Multiplier research whereas authors in [**59**] [**60**] and [**61**] have proposed taxonomy designs reflecting 5, 6, and 14 levels respectively. Use of CSF and other algorithms have been first discussed in [**48**]. The benefit of using CSF is that it forms the elementary algorithm for other algorithms for generation of taxonomies.

3.2 TAXONOMY GENERATION PROCESS

The taxonomy has been generated in three steps, of which the first and the third steps have been implemented using python 3.7 codes on Windows 10 having 8 GB DDR3 memory, and the second step is manual.

3.2.1 STEP 1: VOCABULARY GENERATION

Generation of taxonomy mandates selection of an indexing and abstracting database. In this paper the database of choice has been IEEE Xplore. Although multiple other databases exits such as SCOPUS, the authors have concentrated on IEEE Xplore as there exists an option where the complete bibliometric information can be downloaded as a comma separated (CSV) file from IEEE Xplore. This file forms the basis for data mining. The CSV file has been named as 'exportfile' and comprises of 29 attributes as mentioned in Table 3.1. The data in 'exportfile' is generally termed as '*Raw Data*'. Table 3.2 presents the total number of articles extracted from IEEE Xplore using the seed term '*QCA*' and '*Quantum Dot Cellular Automata*'. The research in this paper has been done using data extracted from IEEE Xplore on 12th July 2019. Table 3.2 reflects that 751 unique articles comprised the vocabulary file 'exportfile'. Although IEEE Xplore provides articles of seven subtypes, viz. Conferences, journals, Magazines, Early Access Articles, Courses, Books, and Standards, the authors concentrated on conferences and journals only as they reflect the comprehensive research trend.

TABLE 3.1. ATTRIBUTES OF 'EXPORTFILE'

Attributes	Attributes
Document Title	PDF Link
Authors	Author Keywords
Author Affiliations	IEEE Terms
Publication Title	INSPEC Controlled Terms
Date Added To Xplore	INSPEC Non-Controlled Terms
Publication_Year	Mesh_Terms
Volume	Article Citation Count
Issue	Reference Count
Start Page	License
End Page	Online Date
Abstract	Issue Date
ISSN	Meeting Date
ISBNs	Publisher
DOI	Document Identifier
Funding Information	

TABLE 3.2. ARTICLE COUNT

Seed Term	Article Count
QCA	699
Quantum Dot Cellular Automata	609
Total	1308
Duplicates	557
Unique articles	751

Out of the 29 attributes in ‘exportfile’, for the sake of taxonomy, only six attributes were considered; these were Document Title, Authors, Publication Year, Abstract, Author Keywords, and Document type. A new CSV file named ‘metadata’ was generated from ‘exportfile’ containing the six attributes with an addition attribute ‘PiD’. The ‘metadata’ file contained data for all the 751 articles pertaining to the six attributes and consecutive PiD numbers. The PiD number is basically the paper id assigned to each article. The PiD is mapped to the article as shown in Equation 2.

$$PiD_i = [\mathbf{62}]\text{⍰}\rightarrow i \; \forall \, i \;\; \in [1{,}751] \; (2)$$

After generation of ‘metadata’ file, three further databases are generated from ‘metadata’ – ‘keyword_file’, ‘abstract_file’ and ‘research_growth_file’. Table 3.3, Table 3.4, and Table 3.5 provide the schema for the three databases.

TABLE 3.3. SCHEMA FOR ‘KEYWORD_FILE’

Term Id	Author Keyword

TABLE 3.4. SCHEMA FOR ‘ABSTRACT_FILE’

Paper Id	Abstract

TABLE 3.5. SCHEMA FOR ‘RESEARCH_GROWTH_FILE’

Year	Journal #	Conference #	Total

3.2.2 STEP 2: MANUAL TAXONOMY NODE GENERATION

This step generates the Taxonomy nodes. Basically the taxonomy nodes are the final AKs shortlisted after eliminating redundancies and clubbing of relevant AKs.

The 'keyword_file' contains all the unique AKs extracted from all the 751 articles. These AKs are provided with a Term Id. A total of 1017 AKs were extracted. After removal of redundancies, multiple similar/related AKs were found which were clubbed into unique terms, which later formed the nodes of the taxonomy. Table 6 presents some of the major terms along with the AKs associated with them. The final count of such unique terms was observed to be 21. These 21 terms formed the nodes of the taxonomy. The 'abstract_file' contains the abstracts for all the 751 articles. The 'research_growth_file' contains the journal and conference count per year starting from 1985 till 2018 for research on QCA.

TABLE 3.6. UNIQUE TERMS AND ASSOCIATED AKS

Term	Author Keywords
QCA	Quantumdot, Quantumdot Cellular Automata, Quantumdot Cellular Automata QCA, Quantum Dot, Quantum Dot Cellular Automata, Quantum Cellular Automata, Quantum Dot Cellular Automata, QCA, Quantum Cellular Automata QCA, Quantum Dots, QCA Quantumdot Cellular Automata
QCA Technology	Emerging Technologies, Computing, Quantum Computing, Emerging Technology
QCA Architecture	Architecture, Coplanar, Tunnelling, QCA Devices, Multilayer
QCA Types	Nanomagnets, molecular QCA, Quasicrystalline Approximation QCA

Nanoelectronics	Nanoelectronics, Nanotechnologies, Nanotechnology
Digital Circuits	Arithmetic, Adder, Adders, Multiplier, Ripple Carry, Full Adder
Combinational	Inverters, Parity, XOR, NOT, Multiplexer, Comparator
Sequential Circuits	Memory, Register, Latch, Feedback
Fault Tolerance	Tolerance, Fault, Defect Tolerance
Logic Gates	Gate, Gates, Majority Gates, Majority Gate, Logical, Majority Voter, Logic Design, Logic Gates, Majority Logic, Majority, Logic Gate, Logic Synthesis
Modelling and Simulation	Modelling, QCA Designer Tool, QCA Designer, Simulation, QCA Designer, Interactions, Polarization
Power & energy	Low Power, Power Dissipation, Low Power Consumption, Power Consumption, Energy Dissipation
Layout	Density, Area, Scaling, Set, Layout, Temperature, Routing, Testing, Latency, Timing, Scattering, Cell, Cell Count
Circuit Design	Generator, Defect, Quantum Cost, Manufacturing, QCA Circuits
Clocking	Clock, Delay
Miscellaneous	Dense Media Radiative Transfer, Angi-

	ography, Microwave, Probability, Information, Communication, Correlation

Table 3.7 and Figure 3.1 present the research growth in the last 25 years in the domain of QCA. Table 3.7 is the complete dataset for schema provided in Table 3.5.

TABLE 3.7. RESEARCH GROWTH IN QCA

Year	Journal #	Conference #	Total
1985	1	0	1
1989	0	1	1
1990	0	1	1
1992	0	1	1
1993	0	3	3
1994	0	3	3
1995	0	3	3
1996	1	2	3
1997	1	2	3
1998	0	5	5
1999	0	7	7
2000	3	9	12
2001	3	16	19
2002	3	8	11
2003	5	15	20
2004	10	29	39
2005	6	26	32
2006	5	34	39
2007	8	25	33
2008	8	49	57
2009	5	32	37
2010	6	28	34
2011	15	28	43

2012	10	36	46
2013	11	25	36
2014	9	34	43
2015	14	38	52
2016	7	54	61
2017	9	43	52
2018	12	27	39

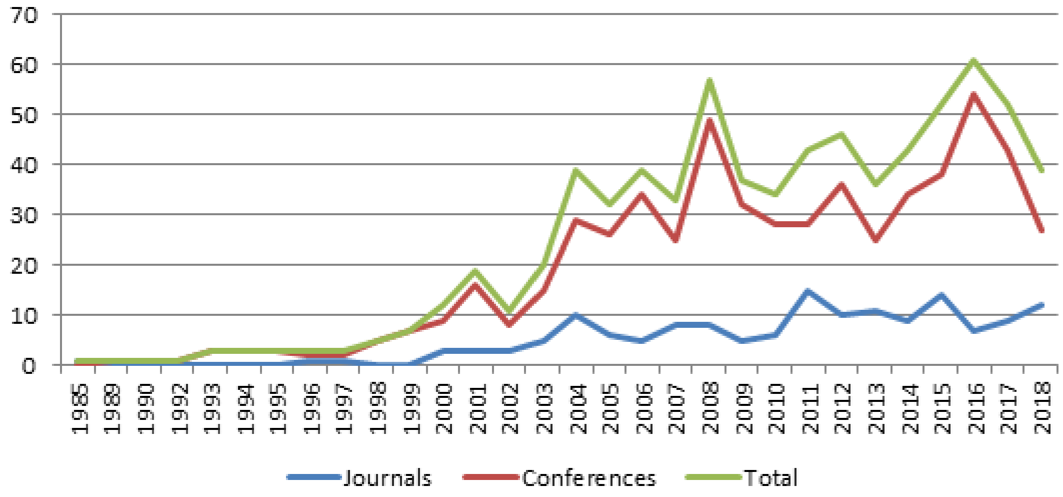


Figure 3.1. Growth of research in the domain of QCA

From Figure 3.1, it can be deduced that the research growth in QCA has been constantly increasing with highest research observed in the previous decade, majorly in the past five years. Also the major contribution has been gained from conference publications. A minor dip in interest has been observed in 2018, but that can be attributed in delay in publications.

Post generation of the 'keyword_file' and 'abstract_file', the fourth database is generated. The authors name it the 'index' database. The 'index' database is generated from the 'keyword_file' and 'abstract_file'. Table 3.8 provides the schema for the 'index' database.

TABLE 3.8. SCHEMA FOR 'INDEX'

AK	PiD	Frequency

This database contains all the PiDs which have a certain AK. The Frequency attribute is the PiD count for a certain AK. The 'index' database is sorted in descending order to have the most occurring AK at the top. This forms the root node of the taxonomy. Readers are directed to Appendix I, Table 3.13 for complete list of AKs and their occurring PiDs.

Table 3.8 presents the taxonomy nodes (AKs) along with frequencies of occurrences in articles.

TABLE 3.9. TAXONOMY NODE (AK) AND RESPECTIVE FREQUENCY

Taxonomy node (AK)	Frequency	Taxonomy node (AK)	Frequency
QCA	687	Power & Energy	178
Layout	463	Digital Circuits	142
Logic Design	304	Miscellaneous	170
Modelling & Simulation	267	Sequential Circuits	90
QCA Technology	231	Fault Tolerance	75
Circuit Design	212	QCA Types	72
Clocking	178	Reversible Logic	41
QCA Architecture	205	Reliability	28
CMOS Technology	196	Optimization	28
Nanoelectronics	189	FPGA	16
Combinational	185		

As discussed earlier, Table 3.9 has been sorted in descending order to have the highest frequency node at the top. This node forms the root node. Hence, in this case, QCA forms the root node and that should be the case as the taxonomy strives to provide research dimensions for research on QCA.

3.2.3 STEP – 3: TAXONOMY GENERATION

The preliminary step in the taxonomy generation process is the generation of the *Term-Occurrence Matrix*. It is generated as per the numerator in Equation (1), i.e. the article count for the occurrence of two AKs simultaneously. Table 3.10 provides the *Term-Occurrence Matrix*. Each cell presents the document count for presence of two AKs denoted by the row and the column heading, in terms of AKi (Term Id of Schema in Table 3.3). It can be observed that the matrix reflects mirror values across the diagonal.

Having generated the *Term-Occurrence Matrix*, the next step is to implement the denominator in accordance to Equation (1). The matrix is shown in Table 3.11. The values in the cells of Table 3.11 have been rounded off to nearest integer value.

The next step is to generate the matrix for implementing the Cosine Similarity function in accordance to Equation (1). Table 3.12 presents the Cosine Similarity values for between two AKs/taxonomy nodes.

Cosine Similarity function has been used because the angle between two nodes denotes direct proportionality between their similarity index. Therefore, if the angle is "0°", then the two nodes are exactly similar, and if the nodes are orthogonal, then $Cos\ 90° = 0 \Rightarrow$completely dissimilar. Hence, the values range between 0 and 1 for all the combinations. Table 3.12 presents the Cosine Similarity Matrix.

TABLE 3.10. TERM-OCCURRENCE MATRIX

PART – I

R/C	AK1	AK2	AK3	AK4	AK5	AK6	AK7	AK8	AK9	AK10	AK11
AK1	-	433	286	248	226	209	195	188	179	173	172
AK2	433	-	212	183	143	144	141	142	106	129	139
AK3	286	212	-	126	104	94	74	100	98	106	77
AK4	248	183	126	-	80	90	74	78	66	75	64
AK5	226	143	104	80	-	88	71	81	60	67	65
AK6	209	144	94	90	88	-	66	55	68	63	59
AK7	195	141	74	74	71	66	-	53	49	41	59
AK8	188	142	100	78	81	55	53	-	71	59	58
AK9	179	106	98	66	60	68	49	71	-	51	39
AK10	173	129	106	75	67	63	41	59	51	-	55
AK11	172	139	77	64	65	59	59	58	39	55	-
AK12	171	131	96	74	85	66	47	80	61	57	63
AK13	142	108	38	61	39	42	37	37	38	47	36
AK14	139	103	83	58	51	46	40	43	45	35	62
AK15	88	67	34	32	30	38	37	36	20	16	37
AK16	73	56	39	26	29	54	23	15	28	18	14
AK17	70	51	16	25	21	12	22	11	9	14	15
AK18	38	29	37	11	23	20	7	14	18	17	14
AK19	23	12	16	9	10	10	7	6	14	6	2
AK20	26	14	18	8	11	9	8	9	5	8	12
AK21	16	11	12	12	10	5	4	8	8	3	1

PART – II

R/C	AK12	AK13	AK14	AK15	AK16	AK17	AK18	AK19	AK20	AK21
AK1	171	142	139	88	73	70	38	23	26	16
AK2	131	108	103	67	56	51	29	12	14	11
AK3	96	38	83	34	39	16	37	16	18	12
AK4	74	61	58	32	26	25	11	9	8	12
AK5	85	39	51	30	29	21	23	10	11	10
AK6	66	42	46	38	54	12	20	10	9	5
AK7	47	37	40	37	23	22	7	7	8	4
AK8	80	37	43	36	15	11	14	6	9	8
AK9	61	38	45	20	28	9	18	14	5	8
AK10	57	47	35	16	18	14	17	6	8	3
AK11	63	36	62	37	14	15	14	2	12	1
AK12	-	39	43	28	13	13	28	7	6	1
AK13	39	-	19	19	5	29	11	8	5	3
AK14	43	19	-	11	13	3	12	3	4	1
AK15	28	19	11	-	11	12	8	0	3	2
AK16	13	5	13	11	-	7	8	4	2	1
AK17	13	29	3	12	7	-	2	2	0	1
AK18	28	11	12	8	8	2	-	2	3	1
AK19	7	8	3	0	4	2	2	-	0	0
AK20	6	5	4	3	2	0	3	0	-	0
AK21	1	3	1	2	1	1	1	0	0	-

TABLE 3.11. DENOMINATOR FOR EQUATION (1)

PART – I

R/C	AK1	AK2	AK3	AK4	AK5	AK6	AK7	AK8	AK9	AK10	AK11
AK1	-	564	457	428	398	382	375	367	360	357	350
AK2	564	-	375	352	327	313	308	301	296	293	287
AK3	457	375	-	285	265	254	250	244	240	237	233
AK4	428	352	285	-	248	238	234	229	225	222	218
AK5	398	327	265	248	-	221	218	213	209	207	203
AK6	382	313	254	238	221	-	208	204	200	198	194
AK7	375	308	250	234	218	208	-	200	197	195	191
AK8	367	301	244	229	213	204	200	-	192	190	187
AK9	360	296	240	225	209	200	197	192	-	187	183
AK10	357	293	237	222	207	198	195	190	187	-	181
AK11	350	287	233	218	203	194	191	187	183	181	-
AK12	350	287	233	218	203	194	191	187	183	181	178
AK13	342	281	227	213	198	190	187	183	179	177	174
AK14	312	256	208	195	181	174	171	167	164	162	159
AK15	249	204	165	155	144	138	136	133	130	129	127
AK16	227	186	151	142	132	126	124	121	119	118	116
AK17	222	183	148	139	129	124	121	119	117	115	113
AK18	168	138	112	105	97	93	92	90	88	87	85
AK19	139	114	92	86	80	77	76	74	73	72	71
AK20	139	114	92	86	80	77	76	74	73	72	71
AK21	105	86	70	65	61	58	57	56	55	54	53

PART – II

R/C	AK12	AK13	AK14	AK15	AK16	AK17	AK18	AK19	AK20	AK21
AK1	350	342	312	249	227	222	168	139	139	105
AK2	287	281	256	204	186	183	138	114	114	86
AK3	233	227	208	165	151	148	112	92	92	70
AK4	218	213	195	155	142	139	105	86	86	65
AK5	203	198	181	144	132	129	97	80	80	61
AK6	194	190	174	138	126	124	93	77	77	58
AK7	191	187	171	136	124	121	92	76	76	57
AK8	187	183	167	133	121	119	90	74	74	56
AK9	183	179	164	130	119	117	88	73	73	55
AK10	181	177	162	129	118	115	87	72	72	54
AK11	178	174	159	127	116	113	85	71	71	53
AK12	-	174	159	127	116	113	85	71	71	53
AK13	174	-	155	124	113	111	83	69	69	52
AK14	159	155	-	113	103	101	76	63	63	48
AK15	127	124	113	-	82	81	61	50	50	38
AK16	116	113	103	82	-	73	55	46	46	35
AK17	113	111	101	81	73	-	54	45	45	34
AK18	85	83	76	61	55	54	-	34	34	26
AK19	71	69	63	50	46	45	34	-	28	21
AK20	71	69	63	50	46	45	34	28	-	21
AK21	53	52	48	38	35	34	26	21	21	-

TABLE 3.12. COSINE SIMILARITY MATRIX

PART – I

R/C	AK1	AK2	AK3	AK4	AK5	AK6	AK7	AK8	AK9	AK10	AK11
AK1	-	0.77	0.63	0.58	0.57	0.55	0.52	0.51	0.5	0.49	0.49
AK2	0.77	-	0.57	0.52	0.44	0.46	0.46	0.47	0.36	0.44	0.48
AK3	0.63	0.57	-	0.44	0.39	0.37	0.3	0.41	0.41	0.45	0.33
AK4	0.58	0.52	0.44	-	0.32	0.38	0.32	0.34	0.29	0.34	0.29
AK5	0.57	0.44	0.39	0.32	-	0.4	0.33	0.38	0.29	0.32	0.32
AK6	0.55	0.46	0.37	0.38	0.4	-	0.32	0.27	0.34	0.32	0.3
AK7	0.52	0.46	0.3	0.32	0.33	0.32	-	0.26	0.25	0.21	0.31
AK8	0.51	0.47	0.41	0.34	0.38	0.27	0.26	-	0.37	0.31	0.31
AK9	0.5	0.36	0.41	0.29	0.29	0.34	0.25	0.37	-	0.27	0.21
AK10	0.49	0.44	0.45	0.34	0.32	0.32	0.21	0.31	0.27	-	0.3
AK11	0.49	0.48	0.33	0.29	0.32	0.3	0.31	0.31	0.21	0.3	-
AK12	0.49	0.46	0.41	0.34	0.42	0.34	0.25	0.43	0.33	0.31	0.35
AK13	0.42	0.38	0.17	0.29	0.2	0.22	0.2	0.2	0.21	0.27	0.21
AK14	0.45	0.4	0.4	0.3	0.28	0.27	0.23	0.26	0.27	0.22	0.39
AK15	0.35	0.33	0.21	0.21	0.21	0.28	0.27	0.27	0.15	0.12	0.29
AK16	0.32	0.3	0.26	0.18	0.22	0.43	0.19	0.12	0.24	0.15	0.12
AK17	0.31	0.28	0.11	0.18	0.16	0.1	0.18	0.09	0.08	0.12	0.13
AK18	0.23	0.21	0.33	0.11	0.24	0.21	0.08	0.16	0.2	0.2	0.16
AK19	0.17	0.11	0.17	0.1	0.12	0.13	0.09	0.08	0.19	0.08	0.03
AK20	0.19	0.12	0.2	0.09	0.14	0.12	0.11	0.12	0.07	0.11	0.17
AK21	0.15	0.13	0.17	0.18	0.16	0.09	0.07	0.14	0.15	0.06	0.02

PART – II

R/C	AK12	AK13	AK14	AK15	AK16	AK17	AK18	AK19	AK20	AK21
AK1	0.49	0.42	0.45	0.35	0.32	0.31	0.23	0.17	0.19	0.15
AK2	0.46	0.38	0.4	0.33	0.3	0.28	0.21	0.11	0.12	0.13
AK3	0.41	0.17	0.4	0.21	0.26	0.11	0.33	0.17	0.2	0.17
AK4	0.34	0.29	0.3	0.21	0.18	0.18	0.11	0.1	0.09	0.18
AK5	0.42	0.2	0.28	0.21	0.22	0.16	0.24	0.12	0.14	0.16
AK6	0.34	0.22	0.27	0.28	0.43	0.1	0.21	0.13	0.12	0.09
AK7	0.25	0.2	0.23	0.27	0.19	0.18	0.08	0.09	0.11	0.07
AK8	0.43	0.2	0.26	0.27	0.12	0.09	0.16	0.08	0.12	0.14
AK9	0.33	0.21	0.27	0.15	0.24	0.08	0.2	0.19	0.07	0.15
AK10	0.31	0.27	0.22	0.12	0.15	0.12	0.2	0.08	0.11	0.06
AK11	0.35	0.21	0.39	0.29	0.12	0.13	0.16	0.03	0.17	0.02
AK12	-	0.22	0.27	0.22	0.11	0.11	0.33	0.1	0.08	0.02
AK13	0.22	-	0.12	0.15	0.04	0.26	0.13	0.12	0.07	0.06
AK14	0.27	0.12	-	0.1	0.13	0.03	0.16	0.05	0.06	0.02
AK15	0.22	0.15	0.1	-	0.13	0.15	0.13	0	0.06	0.05
AK16	0.11	0.04	0.13	0.13	-	0.1	0.14	0.09	0.04	0.03
AK17	0.11	0.26	0.03	0.15	0.1	-	0.04	0.04	0	0.03
AK18	0.33	0.13	0.16	0.13	0.14	0.04	-	0.06	0.09	0.04
AK19	0.1	0.12	0.05	0	0.09	0.04	0.06	-	0	0
AK20	0.08	0.07	0.06	0.06	0.04	0	0.09	0	-	0
AK21	0.02	0.06	0.02	0.05	0.03	0.03	0.04	0	0	-

The final step is taxonomy generation from Table 3.12. This is done by the following pseudo code.

Let the row index be 'i' and the column index be 'j'. Hence, R[i] stands for the i^{th} row and $C[j]$ stands for the j^{th} column. $\therefore R[i]C[i] \forall i \in [1,21]$ generate no value and have been denoted by '-' in Table 12. Also it is to be noted that AK1 is the root node in the taxonomy since the '*Term- Occurrence Matrix*' has been sorted in descending order as previously discussed.

The pseudo code for taxonomy generation is as follows:

```
Begin:
L1  for $i \in [1,49]$
L2      for $j \in [1,49]$
L3          scan $R[i]C[j]$ for the highest value $\forall\ i \neq x \cap j \neq y$
L4      $x = i$ for highest $R[i]C[j]$
L5      $y = j$ for highest $R[i]C[j]$
L6      if $(AK[y]$is not related to $AK[x]$previously)
L7          $AK[x]$is the child of $AK[y]$
L8      else
L9          goto L2
End
```

The rows are scammed from left to right. The AKi providing the highest value for AKj sis assigned the parent of AKj. The scanning results in following hierarchy.

1. AK1
    a. AK2
    b. AK3
        i. AK18
        ii. AK20
    c. AK4
        i. AK21
    d. AK5
    e. AK6
        i. AK16
    f. AK7
    g. AK8
    h. AK9
        i. AK19
    i. AK10
    j. AK11
    k. AK12
    l. AK13
    m. AK14
    n. AK15
    o. AK17

Replacing the AKs in place of the AKs, the following taxonomy is achieved. Figure 3.2 presents the schematic for the taxonomy on research on QCA.

1. QCA
    - a. Layout
    - b. Logic Design
        - i. Reversible Logic
        - ii. Optimization
    - c. Modelling & Simulation
        - i. FPGA
    - d. QCA Technology
    - e. Circuit Design
        - i. Fault Tolerance
    - f. QCA Architecture
    - g. CMOS Technology
    - h. Nanoelectronics
        - i. Reliability
    - i. Combinational
    - j. Clocking
    - k. Power & Energy
    - l. Miscellaneous
    - m. Digital Circuits
    - n. Sequential Circuits
    - o. QCA Types

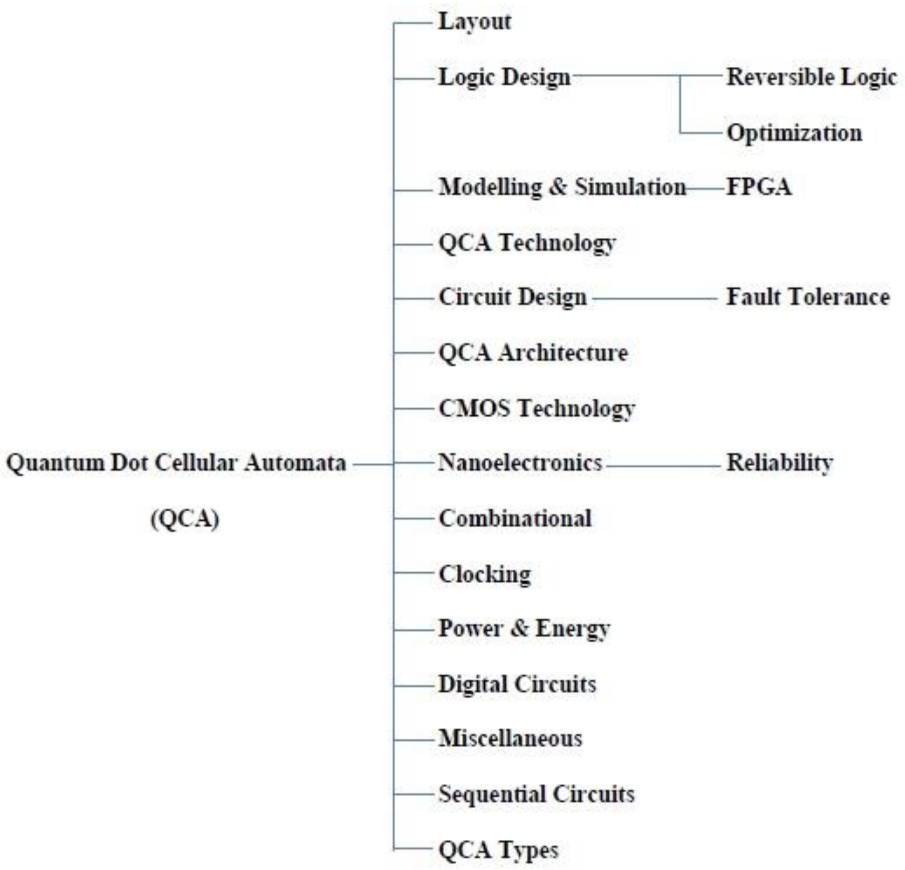


Figure 3.2. Taxonomy for research on QCA

Appendix I provides the literature associated with each taxonomy node. This is one of the major motivations for the research so that a researcher can get a first-hand reference for the node of interest. Appendix II, Table 14 presents the weightage of taxonomy nodes in the articles. The higher is the weightage, more important is the article. It is believed that high weightage articles are a must read as they form the basic foundations for the research.

3.3 TAXONOMY ANALYSIS

It can be observed from Figure 3.2 that the first layer of the taxonomy reflects 15 nodes of which only four nodes have further branches.

It can be further observed that although Logic Design has a frequency of 304, only 41 and 28 articles correlate to Reversible Logic and Optimization respectively. Similarly, 16 proposals have been implemented on FPGAs out of 267 articles based on Modelling & Simulation.

Another observation reveals that out of 212 articles based on Circuit Designs; only 75 articles have dealt with Fault Tolerance. Reliability forms about 15% among proposals involving Nanoelectronics.

The publication domain of QCA majorly concentrates on independent researches within the fifteen first layer nodes. Only five nodes, viz. Reversible Logic, Optimization, FPGA, Fault Tolerance, and Reliability possess pre-requisites in research for QCA. Rest all the dimensions reflect direct connection to QCA.

The major dimensions of research in QCA have been observed in Layout, Logic Design, Modelling & Simulation, QCA Technology, Circuit Design, and QCA architecture contributing 62%, 40%, 36%, 31%, 29%, and 28% respectively among published literature. Although Clocking forms one of the vital dimensions of QCA, it has witnessed about 24% importance with respect to publication count. Research on QCA majorly gained importance due to low power designs in contrast to CMOS technology, but statistics reveal that Power and Energy have contributed only 24% of literature, similar to Clocking.

This thesis covers work related to   nodes in the proposed taxonomy. Chapter 2 corresponds to “reversible Logic” node. Chapter 3 and chapter 4 correspond to the “QCA”, “Digital Circuits” and “QCA Architecture” nodes respectively. Chapter 5 relates to the “Circuit Design” and “Modelling and Simulation” nodes. Chapter 6 corresponds to the “Reversible Logic” and “QCA Architecture” nodes.

3.4 APPENDIX I

This appendix contains all the AKs along with the articles which contain the AK.

TABLE 3.13. LIST OF AKS AND PIDS CONTAINING THE AKS

AK	PiD
QCA	PID1;PID2;PID3;PID4;PID5;PID7;PID8;PID9;PID10;PID11;PID12;PID13;PID14;PID16;PID17;PID18;PID19;PID21;PID23;PID24;PID25;PID26;PID27;PID28;PID29;PID30;PID31;PID33;PID34;PID36;PID37;PID38;PID39;PID40;PID41;PID42;PID43;PID45;PID46;PID47;PID49;PID50;PID52;PID53;PID54;PID56;PID57;PID59;PID60;PID61;PID62;PID63;PID64;PID65;PID66;PID67;PID68;PID69;PID70;PID71;PID72;PID73;PID74;PID75;PID76;PID77;PID78;PID79;PID80;PID82;PID83;PID84;PID85;PID86;PID87;PID88;PID89;PID90;PID91;PID92;PID93;PID94;PID95;PID96;PID97;PID99;PID100;PID101;PID102;PID103;PID104;PID105;PID106;PID108;PID109;PID111;PID112;PID113;PID114;PID116;PID117;PID118;PID120;PID121;PID122;PID123;PID124;PID125;PID126;PID128;PID129;PID130;PID131;PID132;PID133;PID134;PID135;PID136;PID137;PID140;PID141;PID142;PID143;PID146;PID147;PID148;PID149;PID150;PID151;PID152;PID153;PID155;PID156;PID157;PID158;PID159;PID160;PID161;PID163;PID164;PID166;PID167;PID168;PID169;PID170;PID171;PID172;PID173;PID174;PID175;PID176;PID177;PID180;PID181;PID182;PID184;PID185;PID186;PID187;PID188;PID189;PID190;PID191;PID192;PID193;PID194;PID195;PID196;PID197;PID198;PID199;PID200;PID201;PID202;PID203;PID204;PID205;PID206;PID207;PID208;PID209;PID210;PID211;PID212;PID213;PID214;PID215;PID216;PID217;PID218;PID219;PID220;PID221;PID222;PID223;PID224;PID225;PID227;PID228;PID229;PID231;PID232;PID233;PID234;PID235;PID236;PID237;PID238;PID239;PID242;PID245;PID246;PID247;PID248;PID249;PID250;PID251;PID252;PID253;PID254;PID255;PID256;PID258;PID259;PID260;PID261;PID262;PID263;PID264;PID266;PID267;PID268;PID269;PID270;PID271;PID272;PID273;PID274;PID275;PID276;PID277;PID278;PID279;PID280;PID281;PID282;PID283;PID284;PID285;PID286;PID287;PID288;PID289;PID290;PID291;PID292;PID293;PID295;PID296;PID297;PID298;PID299;PID300;PID302;PID303;PID304;PID305;PID306;PID307;PID308;PID309;PID310;PID311;PID312;PID313;PID314;PID315;PID317;PID320;PID321;PID323;PID324;PID325;PID326;PID327;PID330;PID331;PID332;PID335;PID337;PID341;PID342;PID344;PID346;PID347;PID348;PID349;PID350;PID352;PID353;PID354;PID355;PID357;PID359;PID360;PID362;PID363;PID365;PID366;PID367;PID368;PID370;PID373;PID374;PID375;PID376;PID377;PID378;PID379;PID380;PID382;PID383;PID385;PID386;PID387;PID388;PID389;PID390;PID391;PID392;PID393;PID394;PID396;PID397;PID398;PID399;PID400;PID401;PID402;PID403;PID404;PID405;PID407;PID408;PID409;PID411;PID413;PID414;PID415;PID417;PID419;PID421;PID422;PID423;PID424;PID427;PID428;PID431;PID432;PID434;PID436;PID437;PID440;PID441;PID442;PID444;PID445;

	PID446;PID449;PID450;PID455;PID456;PID457;PID458;PID459;PID462;PID465;PID466;PID467;PID468;PID469;PID470;PID471;PID472;PID473;PID475;PID476;PID477;PID478;PID479;PID481;PID482;PID483;PID484;PID489;PID490;PID491;PID492;PID493;PID494;PID495;PID496;PID497;PID498;PID499;PID500;PID501;PID502;PID503;PID504;PID507;PID508;PID509;PID511;PID512;PID513;PID514;PID515;PID516;PID517;PID518;PID521;PID522;PID523;PID524;PID525;PID526;PID527;PID528;PID529;PID530;PID531;PID532;PID533;PID534;PID536;PID537;PID538;PID539;PID541;PID542;PID543;PID545;PID546;PID548;PID549;PID550;PID551;PID552;PID553;PID554;PID555;PID556;PID558;PID559;PID561;PID563;PID564;PID565;PID566;PID567;PID568;PID569;PID570;PID571;PID572;PID573;PID574;PID575;PID576;PID577;PID578;PID580;PID581;PID582;PID583;PID584;PID585;PID586;PID587;PID588;PID589;PID590;PID591;PID592;PID593;PID594;PID595;PID597;PID599;PID600;PID601;PID602;PID603;PID604;PID605;PID607;PID608;PID610;PID611;PID612;PID615;PID616;PID617;PID618;PID619;PID620;PID621;PID622;PID623;PID624;PID625;PID626;PID627;PID630;PID631;PID632;PID634;PID635;PID636;PID637;PID639;PID640;PID641;PID642;PID643;PID645;PID646;PID647;PID648;PID649;PID650;PID651;PID652;PID653;PID654;PID655;PID656;PID657;PID660;PID661;PID662;PID664;PID666;PID668;PID669;PID670;PID671;PID672;PID673;PID674;PID675;PID676;PID677;PID678;PID679;PID681;PID682;PID683;PID684;PID685;PID687;PID690;PID691;PID692;PID693;PID694;PID695;PID696;PID697;PID699;PID700;PID701;PID702;PID708;PID709;PID711;PID713;PID714;PID715;PID716;PID717;PID718;PID720;PID721;PID723;PID725;PID727;PID728;PID729;PID731;PID733;PID734;PID735;PID736;PID737;PID739;PID740;PID741;PID742;PID743;PID744;PID745;PID746;PID747;PID749;PID750;PID751;PID15;PID32;PID35;PID48;PID58;PID119;PID127;PID139;PID162;PID240;PID294;PID361;PID406;PID410;PID418;PID425;PID433;PID453;PID460;PID461;PID463;PID480;PID485;PID519;PID520;PID535;PID547;PID596;PID606;PID638;PID644;PID663;PID665;PID680;PID686;PID688;PID689;PID712;PID722;PID738;PID20;PID55;PID107;PID115;PID178;PID226;PID243;PID257;PID265;PID316;PID351;PID356;PID358;PID384;PID395;PID412;PID464;PID474;PID486;PID487;PID506;PID510;PID560;PID562;PID609;PID710;PID144;PID179;PID230;PID244;PID381;PID598;PID704;PID110;PID183;PID369;PID540;PID138;PID658
Layout	PID4;PID40;PID44;PID46;PID72;PID75;PID94;PID103;PID108;PID137;PID161;PID168;PID179;PID180;PID183;PID187;PID194;PID199;PID204;PID211;PID212;PID220;PID221;PID222;PID228;PID237;PID238;PID240;PID242;PID244;PID253;PID256;PID281;PID287;PID290;PID305;PID309;PID331;PID335;PID341;PID346;PID347;PID354;PID363;PID370;PID392;PID398;PID407;PID415;PID429;PID444;PID450;PID452;PID460;PID462;PID468;PID469;PID481;PID484;PID519;PID524;PID534;PID565;PID575;PID576;PID590;PID591;PID604;PID622;PID646;PID670;PID673;PID706;PID722;PID735;PID740;PID11;PID13;PID16;PID17;PID23;PID24;PID28;PID29;PID52;PID55;PID57;PID60;PID62;PID68;PID69;PID70;PID71;PID87;PID95;PID99;PID116;PID118;PID120;PID121;PID149;PI

	D152;PID155;PID158;PID160;PID174;PID175;PID178;PID188;PID189;PID190;PID191;PID192;PID193;PID195;PID197;PID213;PID219;PID224;PID225;PID226;PID230;PID231;PID233;PID234;PID235;PID239;PID246;PID250;PID252;PID257;PID260;PID264;PID267;PID271;PID272;PID289;PID293;PID295;PID306;PID308;PID312;PID315;PID316;PID349;PID355;PID356;PID357;PID365;PID368;PID379;PID382;PID384;PID385;PID386;PID387;PID388;PID394;PID396;PID400;PID401;PID404;PID408;PID413;PID431;PID438;PID456;PID458;PID492;PID501;PID508;PID520;PID521;PID526;PID529;PID530;PID531;PID532;PID533;PID539;PID548;PID552;PID569;PID570;PID572;PID592;PID593;PID594;PID599;PID605;PID608;PID610;PID611;PID612;PID616;PID623;PID625;PID649;PID653;PID661;PID662;PID675;PID676;PID681;PID686;PID697;PID698;PID700;PID701;PID704;PID707;PID708;PID713;PID723;PID728;PID732;PID733;PID745;PID749;PID129;PID140;PID144;PID173;PID251;PID268;PID292;PID425;PID426;PID493;PID514;PID563;PID582;PID596;PID647;PID650;PID682;PID695;PID49;PID61;PID63;PID65;PID91;PID93;PID102;PID110;PID114;PID115;PID128;PID203;PID245;PID248;PID254;PID263;PID297;PID344;PID376;PID416;PID430;PID432;PID477;PID495;PID522;PID528;PID549;PID550;PID551;PID556;PID584;PID601;PID607;PID626;PID637;PID642;PID665;PID666;PID671;PID690;PID692;PID748;PID35;PID59;PID113;PID124;PID132;PID153;PID218;PID243;PID255;PID259;PID266;PID286;PID298;PID307;PID310;PID374;PID380;PID420;PID443;PID459;PID478;PID489;PID490;PID511;PID513;PID557;PID558;PID567;PID580;PID589;PID597;PID624;PID636;PID687;PID699;PID714;PID717;PID742;PID2;PID100;PID134;PID138;PID143;PID145;PID156;PID169;PID170;PID216;PID326;PID334;PID435;PID448;PID473;PID507;PID535;PID538;PID544;PID553;PID638;PID641;PID648;PID658;PID664;PID685;PID705;PID18;PID176;PID184;PID352;PID375;PID383;PID464;PID41;PID130;PID172;PID362;PID390;PID402;PID518;PID716;PID373;PID691;PID696;PID131;PID217;PID367;PID441;PID446;PID472;PID500;PID504;PID517;PID562;PID603;PID645;PID667;PID1;PID3;PID39;PID43;PID318;PID319;PID320;PID321;PID322;PID324;PID325;PID327;PID328;PID330;PID332;PID337;PID339;PID447;PID451;PID627;PID9;PID10;PID25;PID33;PID112;PID122;PID133;PID135;PID136;PID141;PID151;PID185;PID196;PID198;PID209;PID223;PID229;PID265;PID273;PID283;PID284;PID285;PID348;PID358;PID364;PID369;PID393;PID403;PID411;PID419;PID421;PID423;PID424;PID437;PID449;PID461;PID463;PID476;PID483;PID491;PID497;PID498;PID502;PID503;PID523;PID527;PID540;PID545;PID546;PID547;PID555;PID559;PID587;PID617;PID640;PID651;PID652;PID655;PID657;PID668;PID693;PID702;PID712;PID731;PID736;PID741;PID744;PID746
Logic Design	PID4;PID13;PID15;PID16;PID17;PID19;PID27;PID33;PID37;PID46;PID50;PID68;PID71;PID74;PID76;PID93;PID94;PID95;PID100;PID101;PID103;PID109;PID117;PID118;PID119;PID123;PID133;PID142;PID147;PID152;PID154;PID161;PID168;PID174;PID175;PID176;PID178;PID188;PID190;PID191;PID192;PID193;PID194;PID195;PID199;PID204;PID209;PID215;PID219;PID223;PID226;PID227;PID232;PID233;PID234;PID235;PID238;PID243;PID249;PID251;PID252;PID264;PID265;PID268;PID2

	71;PID274;PID276;PID294;PID295;PID297;PID310;PID312;PID314;PID315;PID355;PID357;PID363;PID365;PID368;PID375;PID377;PID380;PID387;PID390;PID393;PID395;PID396;PID401;PID404;PID410;PID412;PID416;PID417;PID440;PID443;PID455;PID457;PID458;PID463;PID483;PID486;PID487;PID502;PID505;PID509;PID512;PID521;PID522;PID524;PID525;PID528;PID542;PID549;PID557;PID558;PID559;PID560;PID562;PID564;PID567;PID576;PID577;PID578;PID582;PID584;PID587;PID598;PID599;PID601;PID605;PID608;PID611;PID622;PID625;PID643;PID652;PID653;PID667;PID680;PID692;PID697;PID700;PID704;PID708;PID710;PID712;PID713;PID714;PID715;PID718;PID721;PID722;PID731;PID737;PID740;PID741;PID744;PID35;PID58;PID70;PID105;PID114;PID150;PID162;PID239;PID258;PID267;PID388;PID479;PID541;PID613;PID641;PID659;PID664;PID669;PID690;PID701;PID723;PID749;PID25;PID56;PID127;PID134;PID140;PID172;PID202;PID216;PID231;PID246;PID288;PID400;PID407;PID462;PID469;PID495;PID499;PID508;PID519;PID535;PID548;PID555;PID556;PID561;PID589;PID612;PID614;PID649;PID655;PID658;PID673;PID719;PID110;PID129;PID132;PID257;PID385;PID405;PID450;PID624;PID642;PID647;PID716;PID20;PID116;PID121;PID122;PID179;PID244;PID253;PID316;PID411;PID454;PID513;PID518;PID711;PID149;PID196;PID197;PID250;PID260;PID269;PID292;PID435;PID477;PID488;PID571;PID610;PID616;PID678;PID18;PID63;PID342;PID348;PID437;PID529;PID530;PID47;PID65;PID113;PID115;PID125;PID245;PID248;PID311;PID344;PID359;PID389;PID394;PID532;PID688;PID728;PID346;PID565;PID607;PID29;PID32;PID57;PID111;PID145;PID171;PID213;PID256;PID263;PID278;PID289;PID349;PID384;PID397;PID418;PID420;PID434;PID449;PID468;PID501;PID506;PID514;PID544;PID592;PID593;PID657;PID665;PID698;PID732;PID743
Modeling & Simulation	PID48;PID80;PID130;PID200;PID207;PID215;PID229;PID274;PID324;PID331;PID339;PID378;PID456;PID477;PID488;PID491;PID504;PID513;PID559;PID577;PID578;PID641;PID674;PID677;PID692;PID698;PID4;PID16;PID21;PID26;PID100;PID124;PID146;PID174;PID189;PID202;PID209;PID252;PID253;PID268;PID286;PID306;PID368;PID458;PID533;PID548;PID556;PID558;PID612;PID613;PID717;PID739;PID740;PID743;PID11;PID17;PID25;PID29;PID36;PID47;PID59;PID60;PID69;PID71;PID75;PID78;PID95;PID116;PID157;PID208;PID213;PID219;PID223;PID226;PID232;PID272;PID275;PID294;PID298;PID300;PID356;PID357;PID434;PID483;PID501;PID511;PID520;PID526;PID529;PID530;PID534;PID552;PID560;PID567;PID569;PID582;PID609;PID611;PID615;PID619;PID623;PID637;PID643;PID647;PID652;PID664;PID679;PID681;PID693;PID696;PID700;PID708;PID3;PID19;PID24;PID33;PID34;PID41;PID64;PID76;PID102;PID105;PID117;PID118;PID127;PID129;PID131;PID132;PID148;PID150;PID175;PID177;PID185;PID196;PID197;PID210;PID212;PID218;PID231;PID234;PID235;PID238;PID239;PID241;PID250;PID260;PID263;PID269;PID277;PID292;PID293;PID303;PID314;PID315;PID318;PID332;PID335;PID340;PID347;PID353;PID362;PID363;PID370;PID372;PID374;PID375;PID382;PID388;PID397;PID399;PID400;PID401;PID419;PID435;PID443;PID444;PID445;PID449;PID492;PID502;PID507;PID509;PID524;PID535;PID540;PID544;PID545;PID547;PID553;PID557;

	PID563;PID589;PID590;PID602;PID631;PID636;PID646;PID648;PID655;PID657;PID659;PID670;PID672;PID683;PID684;PID687;PID695;PID701;PID714;PID718;PID721;PID731;PID741;PID745;PID746;PID747;PID52;PID62;PID74;PID94;PID134;PID168;PID190;PID228;PID236;PID276;PID295;PID307;PID386;PID455;PID479;PID499;PID515;PID576;PID97;PID112;PID113;PID211;PID220;PID322;PID341;PID430;PID436;PID447;PID468;PID495;PID512;PID603;PID626;PID628;PID638;PID640;PID682;PID706;PID730;PID737;PID14;PID38;PID108;PID352;PID414;PID463;PID476;PID508;PID554;PID651;PID685
QCA Technology	PID9;PID19;PID21;PID47;PID62;PID73;PID78;PID79;PID101;PID111;PID123;PID125;PID129;PID134;PID168;PID174;PID199;PID200;PID208;PID220;PID221;PID228;PID231;PID238;PID239;PID251;PID256;PID267;PID268;PID272;PID275;PID282;PID286;PID292;PID300;PID309;PID360;PID363;PID368;PID376;PID379;PID382;PID392;PID413;PID434;PID456;PID458;PID470;PID517;PID527;PID556;PID563;PID569;PID572;PID576;PID582;PID592;PID594;PID595;PID600;PID604;PID610;PID615;PID625;PID660;PID670;PID673;PID699;PID701;PID720;PID734;PID737;PID153;PID187;PID189;PID192;PID259;PID377;PID384;PID385;PID415;PID440;PID453;PID514;PID522;PID538;PID539;PID570;PID586;PID596;PID597;PID606;PID611;PID675;PID685;PID713;PID722;PID8;PID11;PID16;PID23;PID29;PID42;PID61;PID68;PID74;PID77;PID92;PID93;PID98;PID99;PID102;PID104;PID108;PID110;PID118;PID133;PID146;PID154;PID185;PID191;PID196;PID197;PID205;PID213;PID218;PID222;PID237;PID249;PID258;PID265;PID266;PID277;PID279;PID284;PID285;PID291;PID305;PID308;PID316;PID346;PID347;PID349;PID362;PID369;PID389;PID397;PID402;PID403;PID409;PID412;PID418;PID422;PID425;PID426;PID431;PID441;PID450;PID454;PID455;PID462;PID464;PID469;PID474;PID476;PID478;PID481;PID486;PID489;PID491;PID493;PID496;PID499;PID503;PID506;PID510;PID511;PID518;PID537;PID548;PID554;PID561;PID564;PID573;PID580;PID583;PID593;PID599;PID602;PID608;PID617;PID622;PID642;PID643;PID651;PID655;PID663;PID678;PID683;PID686;PID687;PID691;PID694;PID709;PID710;PID711;PID715;PID716;PID717;PID725;PID729;PID732;PID743;PID30;PID50;PID80;PID103;PID195;PID207;PID226;PID257;PID274;PID278;PID307;PID501;PID571;PID577;PID578;PID669;PID676;PID723
Circuit Design	PID27;PID46;PID53;PID70;PID120;PID130;PID148;PID154;PID159;PID177;PID180;PID258;PID269;PID312;PID349;PID356;PID368;PID374;PID376;PID378;PID385;PID399;PID411;PID413;PID414;PID490;PID491;PID512;PID513;PID517;PID570;PID572;PID594;PID619;PID648;PID655;PID659;PID686;PID699;PID722;PID14;PID16;PID23;PID30;PID34;PID49;PID61;PID64;PID71;PID73;PID77;PID80;PID102;PID108;PID114;PID141;PID149;PID156;PID168;PID169;PID170;PID173;PID185;PID195;PID205;PID207;PID209;PID218;PID221;PID227;PID234;PID246;PID248;PID254;PID259;PID263;PID274;PID285;PID287;PID298;PID306;PID363;PID380;PID382;PID387;PID390;PID403;PID409;PID424;PID473;PID478;PID481;PID489;PID492;PID493;PID495;PID499;PID503;PID511;PID524;PID525;PID531;PID536;PID556;PID559;PID573;PID574;PID576;PID

	577;PID578;PID580;PID581;PID589;PID590;PID591;PID597;PID602;PID605;PID607;PID615;PID636;PID637;PID650;PID651;PID661;PID670;PID677;PID678;PID679;PID687;PID690;PID691;PID709;PID718;PID729;PID736;PID737;PID740;PID744;PID747;PID65;PID118;PID215;PID219;PID354;PID396;PID498;PID537;PID548;PID560;PID692;PID25;PID62;PID112;PID121;PID126;PID129;PID131;PID132;PID151;PID200;PID217;PID348;PID362;PID375;PID392;PID394;PID402;PID410;PID446;PID486;PID516;PID535;PID543;PID558;PID624;PID647;PID652;PID656;PID668;PID697;PID708;PID741;PID20;PID191;PID197;PID260;PID264;PID265;PID384;PID391;PID395;PID397;PID412;PID416;PID586;PID710;PID711;PID715;PID101;PID124;PID125;PID128;PID230;PID273;PID358;PID440;PID444;PID450;PID583;PID646;PID674
QCA Architecture	PID143;PID155;PID223;PID228;PID271;PID403;PID445;PID468;PID507;PID514;PID521;PID524;PID639;PID655;PID673;PID12;PID28;PID56;PID99;PID252;PID298;PID331;PID333;PID336;PID340;PID346;PID349;PID356;PID392;PID398;PID431;PID448;PID526;PID594;PID612;PID626;PID628;PID705;PID707;PID730;PID751;PID9;PID16;PID21;PID26;PID31;PID32;PID42;PID46;PID59;PID72;PID77;PID93;PID96;PID101;PID102;PID137;PID145;PID158;PID173;PID176;PID177;PID183;PID195;PID196;PID203;PID225;PID235;PID246;PID259;PID260;PID264;PID266;PID285;PID299;PID341;PID360;PID367;PID369;PID380;PID391;PID394;PID399;PID408;PID422;PID427;PID434;PID437;PID444;PID449;PID456;PID458;PID461;PID464;PID470;PID475;PID478;PID499;PID503;PID509;PID534;PID535;PID536;PID539;PID551;PID559;PID566;PID567;PID580;PID586;PID591;PID593;PID595;PID597;PID603;PID605;PID618;PID646;PID650;PID654;PID661;PID662;PID663;PID665;PID681;PID686;PID691;PID699;PID700;PID720;PID733;PID735;PID742;PID8;PID13;PID14;PID15;PID69;PID113;PID132;PID168;PID388;PID520;PID552;PID602;PID701;PID709;PID58;PID104;PID109;PID162;PID171;PID215;PID284;PID297;PID344;PID357;PID370;PID423;PID440;PID528;PID598;PID601;PID606;PID616;PID638;PID640;PID680;PID682;PID687;PID688;PID692;PID713;PID748;PID25;PID30;PID120;PID129;PID131;PID140;PID146;PID172;PID206;PID258;PID291;PID362;PID382;PID465;PID466;PID469;PID473;PID490;PID497;PID543;PID545;PID554;PID642;PID647;PID668;PID670;PID671;PID685;PID694;PID717;PID731
CMOS Technology	PID4;PID12;PID17;PID28;PID54;PID68;PID76;PID92;PID99;PID134;PID137;PID146;PID155;PID162;PID168;PID187;PID189;PID192;PID195;PID199;PID220;PID221;PID230;PID238;PID239;PID242;PID248;PID250;PID251;PID258;PID259;PID260;PID266;PID268;PID269;PID272;PID286;PID288;PID292;PID295;PID305;PID308;PID313;PID314;PID315;PID316;PID351;PID355;PID360;PID366;PID376;PID388;PID396;PID413;PID514;PID517;PID528;PID533;PID568;PID569;PID571;PID590;PID594;PID596;PID612;PID615;PID621;PID622;PID623;PID661;PID667;PID678;PID679;PID680;PID701;PID713;PID714;PID720;PID725;PID741;PID742;PID745;PID11;PID30;PID38;PID50;PID64;PID65;PID69;PID70;PID71;PID79;PID93;PID95;PID125;PID140;PID144;PID150;PID151;PID153;PID154;PID161;PID172;PID174;PID177;PID179;PID180;PID183;PID190;PI

	D193;PID194;PID200;PID203;PID204;PID205;PID209;PID211;PID212;PID214;PID223;PID228;PID229;PID233;PID234;PID235;PID244;PID246;PID256;PID261;PID273;PID278;PID281;PID287;PID289;PID290;PID306;PID307;PID347;PID354;PID367;PID370;PID374;PID382;PID383;PID397;PID398;PID405;PID407;PID415;PID417;PID424;PID426;PID435;PID437;PID440;PID454;PID458;PID477;PID494;PID496;PID502;PID504;PID522;PID524;PID536;PID551;PID556;PID558;PID563;PID564;PID565;PID566;PID582;PID584;PID604;PID605;PID610;PID611;PID616;PID646;PID653;PID657;PID660;PID669;PID670;PID673;PID684;PID693;PID698;PID719;PID729;PID732;PID737;PID744;PID746;PID747
Nanoelectronics	PID9;PID72;PID96;PID123;PID126;PID207;PID281;PID290;PID292;PID351;PID356;PID369;PID370;PID388;PID399;PID425;PID460;PID488;PID499;PID505;PID509;PID543;PID545;PID577;PID578;PID597;PID605;PID656;PID659;PID684;PID37;PID183;PID219;PID367;PID421;PID524;PID527;PID576;PID615;PID616;PID748;PID28;PID62;PID63;PID95;PID109;PID114;PID215;PID293;PID344;PID359;PID375;PID383;PID407;PID417;PID437;PID486;PID515;PID516;PID518;PID572;PID598;PID608;PID652;PID688;PID690;PID692;PID698;PID710;PID711;PID723;PID728;PID10;PID12;PID15;PID20;PID24;PID30;PID32;PID36;PID49;PID57;PID60;PID70;PID78;PID79;PID80;PID92;PID107;PID125;PID139;PID148;PID149;PID154;PID155;PID157;PID161;PID186;PID193;PID194;PID195;PID196;PID197;PID198;PID208;PID209;PID210;PID212;PID213;PID230;PID233;PID239;PID246;PID247;PID248;PID250;PID252;PID254;PID261;PID265;PID266;PID269;PID273;PID274;PID275;PID287;PID288;PID289;PID306;PID311;PID314;PID315;PID350;PID365;PID366;PID389;PID396;PID397;PID400;PID402;PID405;PID406;PID410;PID424;PID431;PID439;PID444;PID445;PID453;PID454;PID457;PID459;PID510;PID526;PID533;PID541;PID558;PID564;PID565;PID568;PID569;PID586;PID587;PID591;PID592;PID599;PID607;PID614;PID622;PID624;PID646;PID649;PID662;PID668;PID673;PID715;PID717;PID720;PID721;PID725;PID732;PID734;PID736;PID737;PID738;PID740;PID743;PID745;PID747
Combinational	PID95;PID145;PID190;PID232;PID294;PID359;PID368;PID388;PID479;PID495;PID514;PID525;PID584;PID589;PID643;PID664;PID46;PID154;PID159;PID188;PID195;PID219;PID354;PID375;PID396;PID518;PID537;PID548;PID652;PID673;PID679;PID708;PID743;PID4;PID15;PID17;PID28;PID71;PID123;PID168;PID189;PID204;PID226;PID271;PID293;PID560;PID608;PID740;PID3;PID5;PID9;PID20;PID30;PID31;PID38;PID41;PID44;PID50;PID56;PID65;PID67;PID74;PID81;PID91;PID94;PID111;PID114;PID128;PID134;PID135;PID147;PID149;PID182;PID184;PID206;PID214;PID221;PID245;PID246;PID251;PID254;PID259;PID262;PID267;PID269;PID280;PID285;PID288;PID313;PID317;PID322;PID332;PID352;PID365;PID383;PID412;PID421;PID426;PID439;PID447;PID457;PID463;PID469;PID477;PID478;PID491;PID492;PID493;PID501;PID502;PID504;PID511;PID513;PID524;PID532;PID539;PID541;PID553;PID554;PID570;PID576;PID580;PID590;PID594;PID596;PID597;PID599;PID600;PID620;PID622;PID639;PID640;PID645;PID648;PID653;PID655;PID658;PID665;PID667;PID672;PID678;PID690;PID691;PID695;PID697;PID

	698;PID702;PID710;PID711;PID715;PID718;PID734;PID741;PID748;PID102;PID209;PID222;PID238;PID242;PID256;PID260;PID277;PID282;PID289;PID346;PID393;PID400;PID404;PID431;PID443;PID458;PID612;PID615;PID704;PID11;PID116;PID146;PID160;PID199;PID202;PID265;PID357;PID542;PID551;PID676
Clocking	PID31;PID35;PID47;PID60;PID75;PID93;PID108;PID120;PID124;PID133;PID142;PID160;PID176;PID188;PID204;PID211;PID234;PID242;PID260;PID287;PID295;PID347;PID373;PID383;PID387;PID398;PID424;PID442;PID443;PID478;PID479;PID481;PID490;PID491;PID496;PID498;PID500;PID503;PID517;PID534;PID537;PID550;PID553;PID557;PID566;PID580;PID591;PID600;PID603;PID604;PID619;PID640;PID646;PID651;PID658;PID661;PID663;PID665;PID670;PID682;PID691;PID693;PID695;PID738;PID744;PID10;PID24;PID29;PID44;PID61;PID68;PID95;PID135;PID143;PID147;PID169;PID170;PID192;PID199;PID233;PID259;PID267;PID316;PID374;PID382;PID386;PID392;PID427;PID431;PID434;PID437;PID457;PID466;PID472;PID473;PID502;PID507;PID514;PID521;PID526;PID527;PID533;PID539;PID554;PID555;PID572;PID581;PID582;PID596;PID660;PID666;PID667;PID671;PID673;PID679;PID681;PID685;PID701;PID712;PID720;PID742;PID17;PID20;PID52;PID69;PID106;PID121;PID130;PID149;PID191;PID227;PID231;PID247;PID252;PID255;PID277;PID278;PID293;PID341;PID349;PID354;PID355;PID356;PID385;PID395;PID396;PID397;PID401;PID413;PID415;PID416;PID509;PID520;PID552;PID562;PID586;PID592;PID593;PID594;PID599;PID608;PID609;PID611;PID613;PID649;PID653;PID662;PID664;PID678;PID700;PID710;PID711;PID713;PID715;PID718;PID723;PID741;PID749
Power & Energy	PID4;PID9;PID10;PID38;PID46;PID53;PID62;PID68;PID79;PID99;PID100;PID103;PID111;PID118;PID134;PID154;PID168;PID178;PID187;PID188;PID191;PID195;PID197;PID210;PID212;PID213;PID219;PID226;PID228;PID230;PID231;PID233;PID236;PID238;PID240;PID250;PID251;PID259;PID265;PID266;PID268;PID272;PID278;PID287;PID295;PID307;PID309;PID310;PID316;PID354;PID363;PID383;PID385;PID387;PID389;PID394;PID399;PID406;PID412;PID413;PID424;PID434;PID444;PID455;PID462;PID469;PID514;PID531;PID533;PID537;PID539;PID551;PID560;PID563;PID564;PID565;PID586;PID591;PID597;PID615;PID622;PID646;PID655;PID663;PID667;PID670;PID698;PID709;PID710;PID715;PID719;PID732;PID738;PID745;PID18;PID29;PID74;PID76;PID108;PID135;PID142;PID164;PID169;PID170;PID172;PID194;PID220;PID221;PID235;PID246;PID341;PID366;PID375;PID378;PID460;PID476;PID496;PID502;PID524;PID555;PID562;PID576;PID612;PID617;PID660;PID673;PID744;PID21;PID25;PID57;PID64;PID94;PID192;PID203;PID209;PID217;PID218;PID239;PID248;PID253;PID260;PID267;PID269;PID308;PID347;PID349;PID382;PID401;PID427;PID446;PID459;PID566;PID573;PID590;PID592;PID600;PID608;PID649;PID679;PID685;PID701;PID703;PID720;PID740;PID16;PID37;PID56;PID137;PID140;PID148;PID156;PID190;PID365;PID435;PID540;PID554;PID607;PID625
Digital Circuits	PID4;PID17;PID36;PID47;PID48;PID49;PID52;PID58;PID115;PID119;PID187;PID192;PID196;PID197;PID204;PID231;PID233;PID253;PID266;

	PID293;PID359;PID377;PID379;PID385;PID391;PID416;PID437;PID442;PID499;PID510;PID533;PID548;PID555;PID561;PID575;PID576;PID610;PID611;PID612;PID676;PID678;PID686;PID688;PID718;PID720;PID722;PID740;PID12;PID13;PID20;PID26;PID35;PID57;PID60;PID62;PID69;PID93;PID95;PID102;PID103;PID118;PID120;PID123;PID133;PID161;PID168;PID175;PID178;PID190;PID191;PID213;PID225;PID234;PID252;PID255;PID260;PID264;PID267;PID271;PID273;PID341;PID364;PID366;PID373;PID374;PID380;PID387;PID388;PID394;PID396;PID398;PID410;PID479;PID501;PID512;PID518;PID519;PID526;PID531;PID543;PID552;PID560;PID565;PID571;PID582;PID584;PID592;PID600;PID609;PID613;PID619;PID625;PID643;PID648;PID649;PID651;PID662;PID664;PID671;PID681;PID696;PID717;PID27;PID53;PID230;PID247;PID509;PID529;PID669;PID157;PID376;PID413;PID415;PID459;PID517;PID530;PID536;PID539;PID594;PID608;PID657;PID661
Miscellaneous	PID10;PID20;PID23;PID31;PID33;PID37;PID38;PID44;PID61;PID76;PID81;PID85;PID96;PID118;PID135;PID137;PID139;PID148;PID149;PID184;PID186;PID211;PID212;PID243;PID250;PID251;PID253;PID257;PID260;PID261;PID262;PID266;PID279;PID287;PID291;PID308;PID310;PID314;PID315;PID326;PID338;PID347;PID352;PID371;PID377;PID422;PID423;PID424;PID425;PID431;PID432;PID439;PID449;PID458;PID473;PID477;PID478;PID489;PID494;PID504;PID511;PID513;PID519;PID553;PID554;PID559;PID569;PID580;PID590;PID591;PID605;PID617;PID619;PID630;PID631;PID633;PID634;PID641;PID646;PID662;PID663;PID667;PID675;PID682;PID684;PID685;PID686;PID691;PID695;PID698;PID702;PID706;PID711;PID715;PID732;PID736;PID737;PID743;PID746;PID1;PID2;PID39;PID40;PID321;PID323;PID324;PID325;PID329;PID331;PID334;PID339;PID430;PID448;PID451;PID626;PID705;PID319;PID330;PID332;PID333;PID340;PID429;PID628;PID707;PID730;PID3;PID447;PID43;PID627;PID41;PID66;PID86;PID87;PID88;PID89;PID90;PID91;PID166;PID167;PID270;PID296;PID303;PID34;PID198;PID221;PID491;PID510;PID589;PID648;PID659;PID46;PID48;PID53;PID107;PID119;PID159;PID187;PID195;PID219;PID231;PID354;PID517;PID661;PID679;PID165;PID327;PID328;PID337;PID436;PID588
Sequential Circuits	PID64;PID71;PID131;PID192;PID204;PID223;PID268;PID278;PID349;PID384;PID392;PID395;PID416;PID449;PID513;PID517;PID521;PID564;PID697;PID699;PID710;PID49;PID70;PID158;PID229;PID239;PID257;PID259;PID284;PID285;PID286;PID316;PID381;PID406;PID408;PID411;PID435;PID444;PID494;PID498;PID504;PID534;PID537;PID539;PID545;PID562;PID563;PID567;PID568;PID591;PID621;PID623;PID646;PID663;PID666;PID682;PID687;PID693;PID701;PID744;PID745;PID20;PID136;PID180;PID299;PID347;PID380;PID465;PID466;PID473;PID586;PID618;PID639;PID141;PID394;PID515;PID550;PID558;PID652;PID700;PID23;PID246;PID397;PID398;PID554;PID604;PID661;PID675;PID684;PID702
Fault Tolerance	PID8;PID62;PID126;PID130;PID152;PID154;PID273;PID306;PID308;PID348;PID357;PID364;PID392;PID393;PID394;PID543;PID656;PID675;PID709;PID712;PID112;PID131;PID217;PID362;PID390;PID446;PID516;

	PID535;PID647;PID650;PID741;PID20;PID30;PID65;PID71;PID80;PID110;PID114;PID121;PID122;PID125;PID173;PID200;PID207;PID215;PID254;PID274;PID295;PID349;PID356;PID375;PID391;PID403;PID405;PID415;PID450;PID477;PID518;PID577;PID578;PID625;PID642;PID652;PID668;PID690;PID692;PID697;PID708;PID717;PID23;PID133;PID259;PID385;PID472;PID546
QCA Types	PID75;PID100;PID112;PID123;PID143;PID153;PID206;PID237;PID261;PID285;PID349;PID375;PID476;PID507;PID515;PID538;PID545;PID554;PID583;PID602;PID617;PID652;PID675;PID685;PID687;PID708;PID31;PID127;PID139;PID144;PID145;PID216;PID240;PID351;PID361;PID381;PID480;PID514;PID522;PID535;PID562;PID566;PID660;PID686;PID742;PID1;PID2;PID3;PID39;PID40;PID43;PID321;PID323;PID324;PID325;PID326;PID327;PID330;PID331;PID332;PID337;PID588;PID626;PID35;PID92;PID422;PID435;PID508;PID553;PID654;PID695;PID699
Reversible Logic	PID16;PID20;PID68;PID74;PID99;PID117;PID119;PID121;PID122;PID154;PID178;PID188;PID191;PID195;PID197;PID219;PID222;PID228;PID253;PID260;PID265;PID266;PID278;PID310;PID349;PID365;PID375;PID390;PID395;PID412;PID416;PID518;PID558;PID564;PID565;PID582;PID622;PID710;PID711;PID715;PID732
Reliability	PID15;PID16;PID29;PID31;PID33;PID34;PID123;PID126;PID172;PID221;PID308;PID328;PID405;PID410;PID461;PID486;PID488;PID505;PID510;PID544;PID587;PID622;PID635;PID655;PID656;PID659;PID694;PID747
Optimization	PID50;PID56;PID61;PID84;PID106;PID119;PID164;PID257;PID258;PID277;PID355;PID356;PID357;PID363;PID388;PID395;PID552;PID556;PID590;PID633;PID678;PID680;PID703;PID710;PID713;PID723;PID728;PID749
FPGA	PID101;PID174;PID175;PID282;PID292;PID292;PID358;PID375;PID397;PID407;PID456;PID458;PID567;PID569;PID721;PID737

3.5 APPENDIX II

This appendix contains all the AKs along with the articles which contain the AK.

TABLE 3.14. PERCENTAGE PRESENCE OF AKS (TAXONOMY NODES) IN EACH ARTICLE

PiD	AKs (Taxonomy Nodes)	%	PiD	AKs (Taxonomy Nodes)	%	PiD	AKs (Taxonomy Nodes)	%
PID1	AK2;AK3;AK4;AK5;AK6;AK7;AK8;AK9;AK10;AK11;AK12;AK13;AK14;AK15;AK16;AK17;AK18;AK19;AK20;AK21;AK22	2.06	PID537	AK2;AK6;AK7;AK11;AK12;AK13;AK16	0.69	PID158	AK2;AK3;AK8;AK16	0.39
PID2	AK2;AK3;AK4;AK5;AK6;AK7;AK8;AK9;AK10;AK11;AK12;AK13;AK14;AK15;AK16;AK17;AK18;AK19;AK20;AK21;AK22	2.06	PID545	AK2;AK3;AK5;AK8;AK10;AK16;AK18	0.69	PID159	AK2;AK7;AK11;AK14	0.39
PID3	AK2;AK3;AK4;AK5;AK6;AK7;AK8;AK9;AK10;AK11;AK12;AK13;AK14;AK15;AK16;AK17;AK18;AK19;AK20;AK21;AK22	2.06	PID552	AK2;AK3;AK5;AK8;AK12;AK15;AK21	0.69	PID160	AK2;AK3;AK11;AK12	0.39
PID5	AK2;AK3;AK4;AK5;AK6;AK7;AK8;AK9;AK10;AK11;AK12;AK13;AK14;AK15;AK16;AK17;AK18;AK19;AK20;AK21;AK22	2.06	PID553	AK2;AK3;AK5;AK11;AK12;AK14;AK18	0.69	PID162	AK2;AK4;AK8;AK9	0.39
PID7	AK2;AK3;AK4;AK5;AK6;AK7;AK8;AK9;AK10;AK11;AK12;AK13;AK14;AK15;AK16;AK17;AK18;AK19;AK20;AK21;A	2.06	PID559	AK2;AK3;AK4;AK5;AK7;AK8;AK14	0.69	PID179	AK2;AK3;AK4;AK9	0.39

	K22							
PID4	AK2;AK3;AK4;AK5;AK6;AK7;AK8;AK9;AK10;AK11;AK12;AK13;AK14;AK15;AK16;AK17;AK18;AK19;AK20;AK22	1.97	PID560	AK2;AK4;AK5;AK7;AK11;AK13;AK15	0.69	PID184	AK2;AK3;AK11;AK14	0.39
PID39	AK2;AK3;AK4;AK5;AK6;AK7;AK8;AK9;AK10;AK11;AK12;AK13;AK14;AK15;AK16;AK17;AK18;AK19;AK21;AK22	1.97	PID562	AK2;AK3;AK4;AK12;AK13;AK16;AK18	0.69	PID198	AK2;AK3;AK10;AK14	0.39
PID74	AK2;AK3;AK4;AK5;AK6;AK7;AK8;AK9;AK10;AK11;AK12;AK13;AK14;AK15;AK16;AK17;AK18;AK19;AK20;AK21	1.97	PID563	AK2;AK3;AK5;AK6;AK9;AK13;AK16	0.69	PID202	AK2;AK4;AK5;AK11	0.39
PID34	AK2;AK3;AK4;AK5;AK6;AK7;AK8;AK9;AK10;AK11;AK12;AK13;AK14;AK15;AK16;AK17;AK18;AK19;AK20	1.87	PID567	AK2;AK3;AK4;AK5;AK8;AK16;AK22	0.69	PID205	AK2;AK6;AK7;AK9	0.39
PID56	AK2;AK3;AK4;AK5;AK6;AK7;AK8;AK9;AK10;AK11;AK12;AK13;AK14;AK15;AK16;AK18;AK19;AK21;AK22	1.87	PID577	AK2;AK4;AK5;AK6;AK7;AK10;AK17	0.69	PID206	AK2;AK8;AK11;AK18	0.39
PID15	AK2;AK3;AK4;AK5;AK6;AK7;AK8;AK9;AK10;AK11;AK13;AK14;AK15;AK16;AK17;AK18;AK19;AK20	1.77	PID578	AK2;AK4;AK5;AK6;AK7;AK10;AK17	0.69	PID208	AK2;AK5;AK6;AK10	0.39
PID20	AK2;AK3;AK4;AK5;AK6;AK7;AK8;AK9;AK10;AK11;AK12;AK13;AK14;AK15;AK16;AK1	1.77	PID589	AK2;AK3;AK4;AK5;AK7;AK11;AK14	0.69	PID210	AK2;AK5;AK10;AK13	0.39

	7;AK18;AK19							
PID25	AK2;AK3;AK4;AK5;AK6;AK7;AK8;AK9;AK10;AK11;AK12;AK13;AK14;AK15;AK16;AK17;AK19;AK21	1.77	PID599	AK2;AK3;AK4;AK6;AK10;AK11;AK12	0.69	PID216	AK2;AK3;AK4;AK18	0.39
PID27	AK2;AK3;AK4;AK5;AK6;AK7;AK8;AK9;AK10;AK11;AK12;AK13;AK14;AK15;AK16;AK17;AK19;AK21	1.77	PID625	AK2;AK3;AK4;AK6;AK13;AK15;AK17	0.69	PID225	AK2;AK3;AK8;AK15	0.39
PID29	AK2;AK3;AK4;AK5;AK6;AK7;AK8;AK9;AK10;AK11;AK12;AK13;AK14;AK15;AK16;AK17;AK20;AK22	1.77	PID647	AK2;AK3;AK4;AK5;AK7;AK8;AK17	0.69	PID227	AK2;AK4;AK7;AK12	0.39
PID37	AK2;AK3;AK4;AK5;AK6;AK7;AK8;AK9;AK10;AK11;AK12;AK13;AK14;AK15;AK17;AK18;AK19;AK22	1.77	PID648	AK2;AK3;AK5;AK7;AK11;AK14;AK15	0.69	PID232	AK2;AK4;AK5;AK11	0.39
PID38	AK2;AK3;AK4;AK5;AK6;AK7;AK8;AK9;AK10;AK11;AK12;AK13;AK14;AK15;AK16;AK17;AK18;AK21	1.77	PID649	AK2;AK3;AK4;AK10;AK12;AK13;AK15	0.69	PID237	AK2;AK3;AK6;AK18	0.39
PID40	AK2;AK3;AK4;AK5;AK6;AK7;AK8;AK9;AK10;AK11;AK12;AK13;AK14;AK16;AK17;AK18;AK20;AK22	1.77	PID651	AK2;AK3;AK5;AK6;AK7;AK12;AK15	0.69	PID240	AK2;AK3;AK13;AK18	0.39
PID50	AK2;AK3;AK4;AK5;AK6;AK7;AK8;AK9;AK10;AK11;AK12;AK13;AK14;AK15;AK16;AK18;AK20;AK21	1.77	PID662	AK2;AK3;AK8;AK10;AK12;AK14;AK15	0.69	PID243	AK2;AK3;AK4;AK14	0.39
PID55	AK2;AK3;AK4;AK5;AK6;AK7;AK8;AK9;AK10;AK11;AK12;	1.77	PID663	AK2;AK6;AK8;AK12;AK13;AK14;AK16	0.69	PID244	AK2;AK3;AK4;AK9	0.39

	AK13;AK14;AK15;AK16;AK18;AK19;AK21							
PID58	AK2;AK3;AK4;AK5;AK6;AK7;AK8;AK9;AK10;AK11;AK12;AK13;AK14;AK15;AK16;AK18;AK19;AK20	1.77	PID664	AK2;AK3;AK4;AK5;AK11;AK12;AK15	0.69	PID245	AK2;AK3;AK4;AK11	0.39
PID62	AK2;AK3;AK4;AK5;AK6;AK7;AK8;AK9;AK10;AK11;AK13;AK14;AK15;AK16;AK17;AK18;AK19;AK20	1.77	PID667	AK3;AK4;AK9;AK11;AK12;AK13;AK14	0.69	PID247	AK2;AK10;AK12;AK15	0.39
PID65	AK2;AK3;AK4;AK5;AK6;AK7;AK8;AK9;AK10;AK11;AK12;AK13;AK14;AK15;AK16;AK17;AK18;AK20	1.77	PID675	AK2;AK3;AK6;AK14;AK16;AK17;AK18	0.69	PID255	AK2;AK3;AK12;AK15	0.39
PID67	AK2;AK3;AK4;AK5;AK6;AK7;AK8;AK9;AK10;AK11;AK12;AK13;AK14;AK15;AK16;AK17;AK18;AK21	1.77	PID682	AK2;AK3;AK5;AK8;AK12;AK14;AK16	0.69	PID275	AK2;AK5;AK6;AK10	0.39
PID69	AK2;AK3;AK4;AK5;AK6;AK7;AK8;AK9;AK10;AK11;AK12;AK13;AK14;AK15;AK16;AK17;AK18;AK20	1.77	PID690	AK2;AK3;AK4;AK7;AK10;AK11;AK17	0.69	PID281	AK2;AK3;AK9;AK10	0.39
PID71	AK2;AK3;AK4;AK5;AK6;AK7;AK8;AK9;AK10;AK11;AK12;AK13;AK14;AK15;AK16;AK17;AK19;AK21	1.77	PID695	AK2;AK3;AK5;AK11;AK12;AK14;AK18	0.69	PID282	AK2;AK6;AK11;AK22	0.39
PID11	AK2;AK3;AK4;AK5;AK6;AK7;AK8;AK9;AK10;AK11;AK13;AK14;AK15;AK17;AK18;AK19;AK21	1.67	PID697	AK2;AK3;AK4;AK7;AK11;AK16;AK17	0.69	PID290	AK2;AK3;AK9;AK10	0.39
PID13	AK2;AK3;AK4;AK5;AK6;AK7;	1.67	PID699	AK2;AK3;AK6;AK7;AK	0.69	PID291	AK2;AK6;AK8;AK14	0.39

	AK8;AK9;AK10;AK11;AK12;AK13;AK14;AK15;AK16;AK17;AK18			8;AK16;AK18				
PID16	AK2;AK3;AK4;AK5;AK6;AK7;AK8;AK9;AK10;AK11;AK12;AK13;AK14;AK15;AK19;AK20;AK21	1.67	PID700	AK2;AK3;AK4;AK5;AK8;AK12;AK16	0.69	PID294	AK2;AK4;AK5;AK11	0.39
PID21	AK2;AK3;AK4;AK5;AK6;AK7;AK8;AK9;AK10;AK11;AK12;AK13;AK14;AK15;AK17;AK18;AK19	1.67	PID718	AK2;AK4;AK5;AK7;AK11;AK12;AK15	0.69	PID297	AK2;AK3;AK4;AK8	0.39
PID23	AK2;AK3;AK4;AK5;AK6;AK7;AK8;AK9;AK10;AK11;AK12;AK13;AK14;AK15;AK16;AK17;AK18	1.67	PID723	AK2;AK3;AK4;AK6;AK10;AK12;AK21	0.69	PID305	AK2;AK3;AK6;AK9	0.39
PID26	AK2;AK3;AK4;AK5;AK6;AK7;AK8;AK9;AK10;AK11;AK12;AK13;AK14;AK15;AK16;AK18;AK19	1.67	PID743	AK2;AK4;AK5;AK6;AK10;AK11;AK14	0.69	PID309	AK2;AK3;AK6;AK13	0.39
PID31	AK2;AK3;AK4;AK5;AK6;AK7;AK8;AK9;AK10;AK11;AK12;AK13;AK14;AK16;AK18;AK19;AK20	1.67	PID745	AK2;AK3;AK5;AK9;AK10;AK13;AK16	0.69	PID312	AK2;AK3;AK4;AK7	0.39
PID35	AK2;AK3;AK4;AK5;AK7;AK8;AK9;AK10;AK11;AK12;AK13;AK14;AK15;AK17;AK18;AK21;AK22	1.67	PID75	AK2;AK3;AK5;AK8;AK12;AK18	0.59	PID321	AK2;AK3;AK14;AK18	0.39
PID41	AK2;AK3;AK4;AK5;AK6;AK7;AK9;AK10;AK11;AK12;AK13;AK14;AK15;AK16;AK17;AK19;AK20	1.67	PID76	AK2;AK4;AK5;AK9;AK13;AK14	0.59	PID325	AK2;AK3;AK14;AK18	0.39

PID53	AK2;AK3;AK4;AK5;AK6;AK7;AK8;AK9;AK10;AK11;AK12;AK13;AK14;AK15;AK16;AK17;AK18	1.67	PID80	AK2;AK5;AK6;AK7;AK10;AK17	0.59	PID326	AK2;AK3;AK14;AK18	0.39
PID61	AK2;AK3;AK4;AK5;AK6;AK7;AK8;AK9;AK10;AK11;AK12;AK13;AK14;AK15;AK16;AK18;AK21	1.67	PID94	AK2;AK3;AK4;AK5;AK11;AK13	0.59	PID327	AK2;AK3;AK14;AK18	0.39
PID66	AK2;AK3;AK4;AK5;AK6;AK7;AK8;AK9;AK10;AK11;AK12;AK13;AK14;AK15;AK16;AK17;AK18	1.67	PID100	AK2;AK3;AK4;AK5;AK13;AK18	0.59	PID330	AK2;AK3;AK14;AK18	0.39
PID68	AK2;AK3;AK4;AK5;AK6;AK7;AK8;AK9;AK10;AK12;AK13;AK14;AK15;AK16;AK18;AK19;AK21	1.67	PID101	AK2;AK4;AK6;AK7;AK8;AK22	0.59	PID337	AK2;AK3;AK14;AK18	0.39
PID70	AK2;AK3;AK4;AK5;AK6;AK7;AK8;AK9;AK10;AK11;AK12;AK13;AK14;AK16;AK17;AK18;AK21	1.67	PID103	AK2;AK3;AK4;AK6;AK13;AK15	0.59	PID351	AK2;AK9;AK10;AK18	0.39
PID10	AK2;AK3;AK4;AK5;AK6;AK7;AK8;AK10;AK11;AK12;AK13;AK14;AK15;AK18;AK21;AK22	1.57	PID112	AK2;AK3;AK5;AK7;AK17;AK18	0.59	PID358	AK2;AK3;AK7;AK22	0.39
PID12	AK2;AK3;AK4;AK5;AK6;AK7;AK8;AK9;AK10;AK11;AK12;AK15;AK17;AK18;AK19;AK20	1.57	PID119	AK2;AK4;AK14;AK15;AK19;AK21	0.59	PID360	AK2;AK6;AK8;AK9	0.39
PID17	AK2;AK3;AK4;AK5;AK6;AK7;AK8;AK9;AK11;AK12;AK13;AK15;AK17;A	1.57	PID120	AK2;AK3;AK7;AK8;AK12;AK15	0.59	PID373	AK2;AK3;AK12;AK15	0.39

	K19;AK20;AK22							
PID18	AK2;AK3;AK4;AK5;AK6;AK7;AK8;AK9;AK10;AK11;AK12;AK13;AK14;AK15;AK16;AK19	1.57	PID130	AK2;AK3;AK5;AK7;AK12;AK17	0.59	PID378	AK2;AK5;AK7;AK13	0.39
PID19	AK2;AK3;AK4;AK5;AK6;AK7;AK8;AK9;AK10;AK11;AK12;AK13;AK14;AK15;AK16;AK19	1.57	PID132	AK2;AK3;AK4;AK5;AK7;AK8	0.59	PID379	AK2;AK3;AK6;AK15	0.39
PID24	AK2;AK3;AK4;AK5;AK6;AK7;AK8;AK9;AK10;AK11;AK12;AK13;AK14;AK15;AK16;AK18	1.57	PID135	AK2;AK3;AK11;AK12;AK13;AK14	0.59	PID386	AK2;AK3;AK5;AK12	0.39
PID28	AK2;AK3;AK4;AK5;AK6;AK7;AK8;AK9;AK10;AK11;AK12;AK13;AK14;AK16;AK18;AK22	1.57	PID137	AK2;AK3;AK8;AK9;AK13;AK14	0.59	PID404	AK2;AK3;AK4;AK11	0.39
PID36	AK2;AK3;AK4;AK5;AK6;AK7;AK8;AK9;AK10;AK11;AK13;AK15;AK17;AK18;AK19;AK21	1.57	PID140	AK2;AK3;AK4;AK8;AK9;AK13	0.59	PID406	AK2;AK10;AK13;AK16	0.39
PID44	AK2;AK3;AK4;AK5;AK6;AK7;AK8;AK9;AK10;AK11;AK12;AK13;AK14;AK15;AK16;AK17	1.57	PID146	AK2;AK5;AK6;AK8;AK9;AK11	0.59	PID408	AK2;AK3;AK8;AK16	0.39
PID45	AK2;AK3;AK4;AK5;AK6;AK7;AK8;AK9;AK10;AK11;AK12;AK13;AK14;AK15;AK17;AK22	1.57	PID148	AK2;AK5;AK7;AK10;AK13;AK14	0.59	PID417	AK2;AK4;AK9;AK10	0.39
PID47	AK2;AK3;AK4;AK5;AK6;AK7;AK8;AK9;AK1	1.57	PID161	AK2;AK3;AK4;AK9;AK10;AK15	0.59	PID421	AK2;AK3;AK10;AK11	0.39

	1;AK12;AK13;AK14;AK15;AK16;AK17;AK18							
PID54	AK2;AK3;AK4;AK5;AK6;AK7;AK8;AK9;AK10;AK11;AK13;AK15;AK16;AK17;AK18;AK20	1.57	PID175	AK2;AK3;AK4;AK5;AK15;AK22	0.59	PID423	AK2;AK3;AK8;AK14	0.39
PID59	AK2;AK3;AK4;AK5;AK6;AK7;AK8;AK9;AK10;AK11;AK12;AK13;AK14;AK15;AK16;AK21	1.57	PID178	AK2;AK3;AK4;AK13;AK15;AK19	0.59	PID426	AK3;AK6;AK9;AK11	0.39
PID60	AK2;AK3;AK4;AK5;AK6;AK7;AK8;AK9;AK10;AK11;AK12;AK13;AK14;AK15;AK16;AK18	1.57	PID189	AK2;AK3;AK5;AK6;AK9;AK11	0.59	PID427	AK2;AK8;AK12;AK13	0.39
PID64	AK2;AK3;AK4;AK5;AK6;AK7;AK8;AK9;AK10;AK11;AK12;AK13;AK14;AK15;AK16;AK17	1.57	PID194	AK2;AK3;AK4;AK9;AK10;AK13	0.59	PID445	AK2;AK5;AK8;AK10	0.39
PID9	AK2;AK3;AK4;AK5;AK6;AK8;AK9;AK10;AK11;AK12;AK13;AK14;AK15;AK18;AK19	1.47	PID200	AK2;AK5;AK6;AK7;AK9;AK17	0.59	PID447	AK3;AK5;AK11;AK14	0.39
PID14	AK2;AK3;AK4;AK5;AK6;AK7;AK8;AK9;AK10;AK11;AK12;AK13;AK14;AK16;AK18	1.47	PID207	AK2;AK5;AK6;AK7;AK10;AK17	0.59	PID454	AK4;AK6;AK9;AK10	0.39
PID43	AK2;AK3;AK4;AK5;AK6;AK8;AK9;AK10;AK11;AK12;AK13;AK14;AK15;AK16;AK18	1.47	PID211	AK2;AK3;AK5;AK9;AK12;AK14	0.59	PID460	AK2;AK3;AK10;AK13	0.39
PID49	AK2;AK3;AK4;AK5;AK6;AK7;AK8;AK9;AK10;AK11;AK12;	1.47	PID218	AK2;AK3;AK5;AK6;AK7;AK13	0.59	PID461	AK2;AK3;AK8;AK20	0.39

	AK13;AK14;AK15;AK16							
PID52	AK2;AK3;AK4;AK5;AK6;AK7;AK8;AK9;AK10;AK11;AK12;AK13;AK15;AK16;AK18	1.47	PID220	AK2;AK3;AK5;AK6;AK9;AK13	0.59	PID464	AK2;AK3;AK6;AK8	0.39
PID73	AK2;AK3;AK4;AK5;AK6;AK7;AK8;AK9;AK10;AK11;AK12;AK13;AK14;AK19;AK22	1.47	PID254	AK2;AK3;AK7;AK10;AK11;AK17	0.59	PID466	AK2;AK8;AK12;AK16	0.39
PID46	AK2;AK3;AK4;AK5;AK6;AK7;AK8;AK10;AK11;AK12;AK13;AK14;AK16;AK20	1.38	PID256	AK2;AK3;AK4;AK6;AK9;AK11	0.59	PID472	AK2;AK3;AK12;AK17	0.39
PID72	AK2;AK3;AK4;AK5;AK6;AK7;AK8;AK9;AK10;AK12;AK13;AK15;AK21;AK22	1.38	PID264	AK2;AK3;AK4;AK7;AK8;AK15	0.59	PID483	AK2;AK3;AK4;AK5	0.39
PID30	AK2;AK3;AK5;AK6;AK7;AK8;AK9;AK10;AK11;AK13;AK14;AK17;AK20	1.28	PID271	AK2;AK3;AK4;AK8;AK11;AK15	0.59	PID488	AK4;AK5;AK10;AK20	0.39
PID42	AK2;AK3;AK4;AK6;AK7;AK8;AK9;AK10;AK11;AK12;AK13;AK14;AK18	1.28	PID272	AK2;AK3;AK5;AK6;AK9;AK13	0.59	PID494	AK2;AK9;AK14;AK16	0.39
PID57	AK2;AK3;AK4;AK5;AK6;AK7;AK9;AK10;AK11;AK12;AK13;AK15;AK17	1.28	PID277	AK2;AK5;AK6;AK11;AK12;AK21	0.59	PID516	AK2;AK7;AK10;AK17	0.39
PID260	AK2;AK3;AK4;AK5;AK7;AK8;AK9;AK11;AK12;AK13;AK14;AK15;AK19	1.28	PID286	AK2;AK3;AK5;AK6;AK9;AK16	0.59	PID525	AK2;AK4;AK7;AK11	0.39
PID48	AK2;AK3;AK4;AK5;AK6;AK7;AK10;AK12;AK14;AK15;AK18;AK20	1.18	PID289	AK2;AK3;AK4;AK9;AK10;AK11	0.59	PID532	AK2;AK3;AK4;AK11	0.39
PID63	AK2;AK3;AK4;AK5;AK7;AK8;AK10;AK11;A	1.18	PID307	AK2;AK3;AK5;AK6;AK9;AK13	0.59	PID538	AK2;AK3;AK6;AK18	0.39

	K14;AK16;AK20;AK21							
PID195	AK2;AK3;AK4;AK6;AK7;AK8;AK9;AK10;AK11;AK13;AK14;AK19	1.18	PID310	AK2;AK3;AK4;AK13;AK14;AK19	0.59	PID540	AK2;AK3;AK5;AK13	0.39
PID349	AK2;AK3;AK4;AK6;AK7;AK8;AK12;AK13;AK16;AK17;AK18;AK19	1.18	PID314	AK2;AK4;AK5;AK9;AK10;AK14	0.59	PID541	AK2;AK4;AK10;AK11	0.39
PID375	AK2;AK3;AK4;AK5;AK7;AK10;AK11;AK13;AK17;AK18;AK19;AK22	1.18	PID331	AK2;AK3;AK5;AK8;AK14;AK18	0.59	PID544	AK3;AK4;AK5;AK20	0.39
PID8	AK2;AK3;AK5;AK6;AK7;AK8;AK10;AK11;AK14;AK17;AK21	1.08	PID332	AK2;AK3;AK5;AK11;AK14;AK18	0.59	PID550	AK2;AK3;AK12;AK16	0.39
PID168	AK2;AK3;AK4;AK5;AK6;AK7;AK8;AK9;AK11;AK13;AK15	1.08	PID346	AK2;AK3;AK4;AK6;AK8;AK11	0.59	PID557	AK3;AK4;AK5;AK12	0.39
PID259	AK2;AK3;AK6;AK7;AK8;AK9;AK11;AK12;AK13;AK16;AK17	1.08	PID355	AK2;AK3;AK4;AK9;AK12;AK21	0.59	PID561	AK2;AK4;AK6;AK15	0.39
PID646	AK2;AK3;AK5;AK7;AK8;AK9;AK10;AK12;AK13;AK14;AK16	1.08	PID370	AK2;AK3;AK5;AK8;AK9;AK10	0.59	PID568	AK2;AK9;AK10;AK16	0.39
PID710	AK2;AK4;AK6;AK7;AK10;AK11;AK12;AK13;AK16;AK19;AK21	1.08	PID376	AK2;AK3;AK6;AK7;AK9;AK15	0.59	PID573	AK2;AK6;AK7;AK13	0.39
PID32	AK2;AK3;AK4;AK5;AK8;AK10;AK11;AK14;AK18;AK20	0.98	PID384	AK2;AK3;AK4;AK6;AK7;AK16	0.59	PID583	AK2;AK6;AK7;AK18	0.39
PID197	AK2;AK3;AK4;AK5;AK6;AK7;AK10;AK13;AK15;AK19	0.98	PID390	AK2;AK3;AK4;AK7;AK17;AK19	0.59	PID598	AK2;AK4;AK8;AK10	0.39
PID219	AK2;AK3;AK4;AK5;AK7;AK10;AK11;AK13;AK14;AK19	0.98	PID399	AK2;AK5;AK7;AK8;AK10;AK13	0.59	PID601	AK2;AK3;AK4;AK8	0.39
PID	AK2;AK3;AK4;	0.98	PID4	AK2;AK3;A	0.59	PID6	AK2;AK5;A	0.39

246	AK7;AK8;AK9;AK10;AK11;AK13;AK16		00	K4;AK5;AK10;AK11		09	K12;AK15	
PID266	AK2;AK3;AK6;AK8;AK9;AK10;AK13;AK14;AK15;AK19	0.98	PID401	AK2;AK3;AK4;AK5;AK12;AK13	0.59	PID613	AK4;AK5;AK12;AK15	0.39
PID388	AK2;AK3;AK4;AK5;AK8;AK9;AK10;AK11;AK15;AK21	0.98	PID403	AK2;AK3;AK6;AK7;AK8;AK17	0.59	PID636	AK2;AK3;AK5;AK7	0.39
PID397	AK2;AK4;AK5;AK6;AK7;AK9;AK10;AK12;AK16;AK22	0.98	PID405	AK2;AK4;AK9;AK10;AK17;AK20	0.59	PID637	AK2;AK3;AK5;AK7	0.39
PID458	AK2;AK3;AK4;AK5;AK6;AK8;AK9;AK11;AK14;AK22	0.98	PID407	AK2;AK3;AK4;AK9;AK10;AK22	0.59	PID638	AK2;AK3;AK5;AK8	0.39
PID514	AK2;AK3;AK4;AK6;AK8;AK9;AK11;AK12;AK13;AK18	0.98	PID410	AK2;AK4;AK7;AK10;AK15;AK20	0.59	PID639	AK2;AK8;AK11;AK16	0.39
PID524	AK2;AK3;AK4;AK5;AK7;AK8;AK9;AK10;AK11;AK13	0.98	PID440	AK2;AK4;AK6;AK7;AK8;AK9	0.59	PID666	AK2;AK3;AK12;AK16	0.39
PID554	AK2;AK5;AK6;AK8;AK11;AK12;AK13;AK14;AK16;AK18	0.98	PID450	AK2;AK3;AK4;AK6;AK7;AK17	0.59	PID694	AK2;AK6;AK8;AK20	0.39
PID576	AK2;AK3;AK4;AK5;AK6;AK7;AK10;AK11;AK13;AK15	0.98	PID456	AK2;AK3;AK5;AK6;AK8;AK22	0.59	PID696	AK2;AK3;AK5;AK15	0.39
PID622	AK2;AK3;AK4;AK6;AK9;AK10;AK11;AK13;AK19;AK20	0.98	PID476	AK2;AK3;AK5;AK6;AK13;AK18	0.59	PID704	AK2;AK3;AK4;AK11	0.39
PID652	AK2;AK3;AK4;AK5;AK7;AK10;AK11;AK16;AK17;AK18	0.98	PID479	AK2;AK4;AK5;AK11;AK12;AK15	0.59	PID716	AK2;AK3;AK4;AK6	0.39
PID655	AK2;AK3;AK4;AK5;AK6;AK7;AK8;AK11;AK13;AK20	0.98	PID486	AK2;AK4;AK6;AK7;AK10;AK20	0.59	PID725	AK2;AK6;AK9;AK10	0.39
PID673	AK2;AK3;AK4;AK6;AK8;AK9;AK10;AK11;AK12;AK13	0.98	PID495	AK2;AK3;AK4;AK5;AK7;AK11	0.59	PID729	AK2;AK6;AK7;AK9	0.39
PID701	AK2;AK3;AK4;AK5;AK6;AK8;AK9;AK12;AK	0.98	PID503	AK2;AK3;AK6;AK7;AK8;AK12	0.59	PID734	AK2;AK6;AK10;AK11	0.39

	13;AK16							
PID715	AK2;AK4;AK6;AK7;AK10;AK11;AK12;AK13;AK14;AK19	0.98	PID507	AK2;AK3;AK5;AK8;AK12;AK18	0.59	PID738	AK2;AK10;AK12;AK13	0.39
PID33	AK2;AK3;AK4;AK5;AK8;AK11;AK14;AK18;AK20	0.88	PID510	AK2;AK6;AK10;AK14;AK15;AK20	0.59	PID748	AK3;AK8;AK10;AK11	0.39
PID95	AK2;AK3;AK4;AK5;AK9;AK10;AK11;AK12;AK15	0.88	PID521	AK2;AK3;AK4;AK8;AK12;AK16	0.59	PID87	AK2;AK3;AK14	0.29
PID118	AK2;AK3;AK4;AK5;AK6;AK7;AK13;AK14;AK15	0.88	PID522	AK2;AK3;AK4;AK6;AK9;AK18	0.59	PID104	AK2;AK6;AK8	0.29
PID154	AK4;AK6;AK7;AK9;AK10;AK11;AK13;AK17;AK19	0.88	PID534	AK2;AK3;AK5;AK8;AK12;AK16	0.59	PID105	AK2;AK4;AK5	0.29
PID191	AK2;AK3;AK4;AK6;AK7;AK12;AK13;AK15;AK19	0.88	PID543	AK2;AK7;AK8;AK10;AK15;AK17	0.59	PID106	AK2;AK12;AK21	0.29
PID192	AK2;AK3;AK4;AK6;AK9;AK12;AK13;AK15;AK16	0.88	PID551	AK2;AK3;AK8;AK9;AK11;AK13	0.59	PID107	AK2;AK10;AK14	0.29
PID209	AK2;AK3;AK4;AK5;AK7;AK9;AK10;AK11;AK13	0.88	PID555	AK2;AK3;AK4;AK12;AK13;AK15	0.59	PID136	AK2;AK3;AK16	0.29
PID221	AK2;AK3;AK6;AK7;AK9;AK11;AK13;AK14;AK20	0.88	PID566	AK2;AK8;AK9;AK12;AK13;AK18	0.59	PID164	AK2;AK13;AK21	0.29
PID231	AK2;AK3;AK4;AK5;AK6;AK12;AK13;AK14;AK15	0.88	PID572	AK2;AK3;AK6;AK7;AK10;AK12	0.59	PID171	AK2;AK4;AK8	0.29
PID239	AK2;AK3;AK4;AK5;AK6;AK9;AK10;AK13;AK16	0.88	PID584	AK2;AK3;AK4;AK9;AK11;AK15	0.59	PID186	AK2;AK10;AK14	0.29
PID265	AK2;AK3;AK4;AK6;AK7;AK10;AK11;AK13;AK19	0.88	PID593	AK2;AK3;AK4;AK6;AK8;AK12	0.59	PID214	AK2;AK9;AK11	0.29
PID347	AK2;AK3;AK5;AK6;AK9;AK12;AK13;AK14;AK16	0.88	PID596	AK2;AK3;AK6;AK9;AK11;AK12	0.59	PID236	AK2;AK5;AK13	0.29
PID	AK2;AK3;AK5;	0.88	PID6	AK2;AK6;A	0.59	PID2	AK2;AK4;A	0.29

356	AK7;AK8;AK10;AK12;AK17;AK21		00	K11;AK12;AK13;AK15		49	K6	
PID382	AK2;AK3;AK5;AK6;AK7;AK8;AK9;AK12;AK13	0.88	PID602	AK2;AK5;AK6;AK7;AK8;AK18	0.59	PID262	AK2;AK11;AK14	0.29
PID385	AK2;AK3;AK4;AK6;AK7;AK12;AK13;AK15;AK17	0.88	PID604	AK2;AK3;AK6;AK9;AK12;AK16	0.59	PID276	AK2;AK4;AK5	0.29
PID394	AK2;AK3;AK4;AK7;AK8;AK13;AK15;AK16;AK17	0.88	PID607	AK2;AK3;AK4;AK7;AK10;AK13	0.59	PID279	AK2;AK6;AK14	0.29
PID396	AK2;AK3;AK4;AK7;AK9;AK10;AK11;AK12;AK15	0.88	PID610	AK2;AK3;AK4;AK6;AK9;AK15	0.59	PID299	AK2;AK8;AK16	0.29
PID517	AK2;AK3;AK6;AK7;AK9;AK12;AK14;AK15;AK16	0.88	PID616	AK2;AK3;AK4;AK8;AK9;AK10	0.59	PID300	AK2;AK5;AK6	0.29
PID518	AK2;AK3;AK4;AK6;AK10;AK11;AK15;AK17;AK19	0.88	PID617	AK2;AK3;AK6;AK13;AK14;AK18	0.59	PID303	AK2;AK5;AK14	0.29
PID539	AK2;AK3;AK6;AK8;AK11;AK12;AK13;AK15;AK16	0.88	PID619	AK2;AK5;AK7;AK12;AK14;AK15	0.59	PID311	AK2;AK4;AK10	0.29
PID558	AK2;AK3;AK4;AK5;AK7;AK9;AK10;AK16;AK19	0.88	PID626	AK2;AK3;AK5;AK8;AK14;AK18	0.59	PID313	AK2;AK9;AK11	0.29
PID582	AK2;AK3;AK4;AK5;AK6;AK9;AK12;AK15;AK19	0.88	PID640	AK2;AK3;AK5;AK8;AK11;AK12	0.59	PID322	AK3;AK5;AK11	0.29
PID590	AK2;AK3;AK5;AK7;AK9;AK11;AK13;AK14;AK21	0.88	PID642	AK2;AK3;AK4;AK6;AK8;AK17	0.59	PID323	AK2;AK14;AK18	0.29
PID591	AK2;AK3;AK7;AK8;AK10;AK12;AK13;AK14;AK16	0.88	PID643	AK2;AK4;AK5;AK6;AK11;AK15	0.59	PID328	AK3;AK14;AK20	0.29
PID594	AK2;AK3;AK6;AK7;AK8;AK9;AK11;AK12;AK15	0.88	PID653	AK2;AK3;AK4;AK9;AK11;AK12	0.59	PID335	AK2;AK3;AK5	0.29
PID608	AK2;AK3;AK4;AK6;AK10;AK11;AK12;AK13;	0.88	PID657	AK2;AK3;AK4;AK5;AK9;AK15	0.59	PID339	AK3;AK5;AK14	0.29

	AK15							
PID612	AK2;AK3;AK4;AK5;AK8;AK9;AK11;AK13;AK15	0.88	PID659	AK4;AK5;AK7;AK10;AK14;AK20	0.59	PID340	AK5;AK8;AK14	0.29
PID661	AK2;AK3;AK7;AK8;AK9;AK12;AK14;AK15;AK16	0.88	PID660	AK2;AK6;AK9;AK12;AK13;AK18	0.59	PID364	AK3;AK15;AK17	0.29
PID670	AK2;AK3;AK5;AK6;AK7;AK8;AK9;AK12;AK13	0.88	PID665	AK2;AK3;AK4;AK8;AK11;AK12	0.59	PID381	AK2;AK16;AK18	0.29
PID678	AK2;AK4;AK6;AK7;AK9;AK11;AK12;AK15;AK21	0.88	PID668	AK2;AK3;AK7;AK8;AK10;AK17	0.59	PID409	AK2;AK6;AK7	0.29
PID685	AK2;AK3;AK5;AK6;AK8;AK12;AK13;AK14;AK18	0.88	PID681	AK2;AK3;AK5;AK8;AK12;AK15	0.59	PID414	AK2;AK5;AK7	0.29
PID711	AK2;AK4;AK6;AK7;AK10;AK11;AK12;AK14;AK19	0.88	PID684	AK2;AK5;AK9;AK10;AK14;AK16	0.59	PID418	AK2;AK4;AK6	0.29
PID737	AK2;AK4;AK5;AK6;AK7;AK9;AK10;AK14;AK22	0.88	PID693	AK2;AK3;AK5;AK9;AK12;AK16	0.59	PID419	AK2;AK3;AK5	0.29
PID740	AK2;AK3;AK4;AK5;AK7;AK10;AK11;AK13;AK15	0.88	PID709	AK2;AK6;AK7;AK8;AK13;AK17	0.59	PID430	AK3;AK5;AK14	0.29
PID741	AK2;AK3;AK4;AK5;AK7;AK9;AK11;AK12;AK17	0.88	PID722	AK2;AK3;AK4;AK6;AK7;AK15	0.59	PID432	AK2;AK3;AK14	0.29
PID93	AK2;AK3;AK4;AK6;AK8;AK9;AK12;AK15	0.79	PID742	AK2;AK3;AK8;AK9;AK12;AK18	0.59	PID436	AK2;AK5;AK14	0.29
PID102	AK2;AK3;AK5;AK6;AK7;AK8;AK11;AK15	0.79	PID747	AK2;AK5;AK7;AK9;AK10;AK20	0.59	PID439	AK10;AK11;AK14	0.29
PID123	AK2;AK4;AK6;AK10;AK11;AK15;AK18;AK20	0.79	PID79	AK2;AK6;AK9;AK10;AK13	0.49	PID441	AK2;AK3;AK6	0.29
PID134	AK2;AK3;AK4;AK5;AK6;AK9;AK11;AK13	0.79	PID92	AK2;AK6;AK9;AK10;AK18	0.49	PID442	AK2;AK12;AK15	0.29
PID149	AK2;AK3;AK4;AK7;AK10;AK11;AK12;AK14	0.79	PID110	AK2;AK3;AK4;AK6;AK17	0.49	PID448	AK3;AK8;AK14	0.29
PID	AK2;AK3;AK4;	0.79	PID1	AK2;AK4;A	0.49	PID4	AK2;AK6;A	0.29

190	AK5;AK9;AK11;AK13;AK15		11	K6;AK11;AK13		53	K10	
PID196	AK2;AK3;AK4;AK5;AK6;AK8;AK10;AK15	0.79	PID113	AK2;AK3;AK4;AK5;AK8	0.49	PID465	AK2;AK8;AK16	0.29
PID204	AK2;AK3;AK4;AK9;AK11;AK12;AK15;AK16	0.79	PID116	AK2;AK3;AK4;AK5;AK11	0.49	PID470	AK2;AK6;AK8	0.29
PID213	AK2;AK3;AK4;AK5;AK6;AK10;AK13;AK15	0.79	PID122	AK2;AK3;AK4;AK17;AK19	0.49	PID497	AK2;AK3;AK8	0.29
PID228	AK2;AK3;AK5;AK6;AK8;AK9;AK13;AK19	0.79	PID124	AK2;AK3;AK5;AK7;AK12	0.49	PID500	AK2;AK3;AK12	0.29
PID233	AK2;AK3;AK4;AK9;AK10;AK12;AK13;AK15	0.79	PID126	AK2;AK7;AK10;AK17;AK20	0.49	PID505	AK4;AK10;AK20	0.29
PID234	AK2;AK3;AK4;AK5;AK7;AK9;AK12;AK15	0.79	PID143	AK2;AK3;AK8;AK12;AK18	0.49	PID506	AK2;AK4;AK6	0.29
PID238	AK2;AK3;AK4;AK5;AK6;AK9;AK11;AK13	0.79	PID145	AK3;AK4;AK8;AK11;AK18	0.49	PID542	AK2;AK4;AK11	0.29
PID250	AK2;AK3;AK4;AK5;AK9;AK10;AK13;AK14	0.79	PID153	AK2;AK3;AK6;AK9;AK18	0.49	PID546	AK2;AK3;AK17	0.29
PID251	AK2;AK3;AK4;AK6;AK9;AK11;AK13;AK14	0.79	PID155	AK2;AK3;AK8;AK9;AK10	0.49	PID547	AK2;AK3;AK5	0.29
PID252	AK2;AK3;AK4;AK5;AK8;AK10;AK12;AK15	0.79	PID169	AK2;AK3;AK7;AK12;AK13	0.49	PID549	AK2;AK3;AK4	0.29
PID253	AK2;AK3;AK4;AK5;AK13;AK14;AK15;AK19	0.79	PID170	AK2;AK3;AK7;AK12;AK13	0.49	PID575	AK2;AK3;AK15	0.29
PID267	AK2;AK3;AK4;AK6;AK11;AK12;AK13;AK15	0.79	PID173	AK2;AK3;AK7;AK8;AK17	0.49	PID581	AK2;AK7;AK12	0.29
PID268	AK2;AK3;AK4;AK5;AK6;AK9;AK13;AK16	0.79	PID176	AK2;AK3;AK4;AK8;AK12	0.49	PID588	AK2;AK14;AK18	0.29
PID269	AK2;AK4;AK5;AK7;AK9;AK10;AK11;AK13	0.79	PID177	AK2;AK5;AK7;AK8;AK9	0.49	PID595	AK2;AK6;AK8	0.29
PID278	AK2;AK4;AK6;AK9;AK12;AK13;AK16;AK19	0.79	PID180	AK2;AK3;AK7;AK9;AK16	0.49	PID606	AK2;AK6;AK8	0.29
PID285	AK2;AK3;AK6;AK7;AK8;AK11;AK16;AK18	0.79	PID183	AK2;AK3;AK8;AK9;AK10	0.49	PID618	AK2;AK8;AK16	0.29
PID287	AK2;AK3;AK7;AK9;AK10;AK12;AK13;AK14	0.79	PID185	AK2;AK3;AK5;AK6;AK7	0.49	PID621	AK2;AK9;AK16	0.29
PID	AK2;AK3;AK4;	0.79	PID1	AK2;AK3;A	0.49	PID6	AK2;AK3;A	0.29

292	AK5;AK6;AK9;AK10;AK22		93	K4;AK9;AK10		27	K14	
PID295	AK2;AK3;AK4;AK5;AK9;AK12;AK13;AK17	0.79	PID203	AK2;AK3;AK8;AK9;AK13	0.49	PID628	AK5;AK8;AK14	0.29
PID308	AK2;AK3;AK6;AK9;AK13;AK14;AK17;AK20	0.79	PID217	AK2;AK3;AK7;AK13;AK17	0.49	PID631	AK2;AK5;AK14	0.29
PID316	AK2;AK3;AK4;AK6;AK9;AK12;AK13;AK16	0.79	PID222	AK2;AK3;AK6;AK11;AK19	0.49	PID645	AK2;AK3;AK11	0.29
PID354	AK2;AK3;AK7;AK9;AK11;AK12;AK13;AK14	0.79	PID229	AK2;AK3;AK5;AK9;AK16	0.49	PID654	AK2;AK8;AK18	0.29
PID357	AK2;AK3;AK4;AK5;AK8;AK11;AK17;AK21	0.79	PID242	AK2;AK3;AK9;AK11;AK12	0.49	PID672	AK2;AK5;AK11	0.29
PID363	AK2;AK3;AK4;AK5;AK6;AK7;AK13;AK21	0.79	PID261	AK2;AK9;AK10;AK14;AK18	0.49	PID674	AK2;AK5;AK7	0.29
PID392	AK2;AK3;AK6;AK7;AK8;AK12;AK16;AK17	0.79	PID263	AK2;AK3;AK4;AK5;AK7	0.49	PID677	AK2;AK5;AK7	0.29
PID413	AK2;AK3;AK6;AK7;AK9;AK12;AK13;AK15	0.79	PID284	AK2;AK3;AK6;AK8;AK16	0.49	PID683	AK2;AK5;AK6	0.29
PID424	AK2;AK3;AK7;AK9;AK10;AK12;AK13;AK14	0.79	PID288	AK2;AK4;AK9;AK10;AK11	0.49	PID705	AK3;AK8;AK14	0.29
PID431	AK2;AK3;AK6;AK8;AK10;AK11;AK12;AK14	0.79	PID298	AK2;AK3;AK5;AK7;AK8	0.49	PID706	AK3;AK5;AK14	0.29
PID437	AK2;AK3;AK4;AK8;AK9;AK10;AK12;AK15	0.79	PID324	AK2;AK3;AK5;AK14;AK18	0.49	PID707	AK3;AK8;AK14	0.29
PID444	AK2;AK3;AK5;AK7;AK8;AK10;AK13;AK16	0.79	PID344	AK2;AK3;AK4;AK8;AK10	0.49	PID719	AK4;AK9;AK13	0.29
PID477	AK2;AK3;AK4;AK5;AK9;AK11;AK14;AK17	0.79	PID348	AK2;AK3;AK4;AK7;AK17	0.49	PID730	AK5;AK8;AK14	0.29
PID478	AK2;AK3;AK6;AK7;AK8;AK11;AK12;AK14	0.79	PID352	AK2;AK3;AK5;AK11;AK14	0.49	PID733	AK2;AK3;AK8	0.29
PID491	AK2;AK3;AK5;AK6;AK7;AK11;AK12;AK14	0.79	PID359	AK2;AK4;AK10;AK11;AK15	0.49	PID735	AK2;AK3;AK8	0.29
PID499	AK2;AK4;AK5;AK6;AK7;AK8;AK10;AK15	0.79	PID366	AK2;AK9;AK10;AK13;AK15	0.49	PID81	AK11;AK14	0.20
PID502	AK2;AK3;AK4;AK5;AK9;AK11;AK12;AK13	0.79	PID367	AK2;AK3;AK8;AK9;AK10	0.49	PID84	AK2;AK21	0.20
PID	AK2;AK3;AK4;	0.79	PID3	AK2;AK3;A	0.49	PID8	AK2;AK14	0.20

513	AK5;AK7;AK11;AK14;AK16		69	K6;AK8;AK10		5		
PID533	AK2;AK3;AK5;AK9;AK10;AK12;AK13;AK15	0.79	PID377	AK2;AK4;AK6;AK14;AK15	0.49	PID86	AK2;AK14	0.20
PID535	AK2;AK3;AK4;AK5;AK7;AK8;AK17;AK18	0.79	PID389	AK2;AK4;AK6;AK10;AK13	0.49	PID88	AK2;AK14	0.20
PID548	AK2;AK3;AK4;AK5;AK6;AK7;AK11;AK15	0.79	PID391	AK2;AK7;AK8;AK15;AK17	0.49	PID89	AK2;AK14	0.20
PID556	AK2;AK3;AK4;AK5;AK6;AK7;AK9;AK21	0.79	PID393	AK2;AK3;AK4;AK11;AK17	0.49	PID90	AK2;AK14	0.20
PID564	AK2;AK4;AK6;AK9;AK10;AK13;AK16;AK19	0.79	PID402	AK2;AK3;AK6;AK7;AK10	0.49	PID97	AK2;AK5	0.20
PID565	AK2;AK3;AK4;AK9;AK10;AK13;AK15;AK19	0.79	PID411	AK2;AK3;AK4;AK7;AK16	0.49	PID138	AK2;AK3	0.20
PID569	AK2;AK3;AK5;AK6;AK9;AK10;AK14;AK22	0.79	PID422	AK2;AK6;AK8;AK14;AK18	0.49	PID166	AK2;AK14	0.20
PID580	AK2;AK3;AK6;AK7;AK8;AK11;AK12;AK14	0.79	PID425	AK2;AK3;AK6;AK10;AK14	0.49	PID167	AK2;AK14	0.20
PID586	AK2;AK6;AK7;AK8;AK10;AK12;AK13;AK16	0.79	PID443	AK3;AK4;AK5;AK11;AK12	0.49	PID182	AK2;AK11	0.20
PID592	AK2;AK3;AK4;AK6;AK10;AK12;AK13;AK15	0.79	PID446	AK2;AK3;AK7;AK13;AK17	0.49	PID224	AK2;AK3	0.20
PID597	AK2;AK3;AK6;AK7;AK8;AK10;AK11;AK13	0.79	PID455	AK2;AK4;AK5;AK6;AK13	0.49	PID270	AK2;AK14	0.20
PID605	AK2;AK3;AK4;AK7;AK8;AK9;AK10;AK14	0.79	PID457	AK2;AK4;AK10;AK11;AK12	0.49	PID280	AK2;AK11	0.20
PID611	AK2;AK3;AK4;AK5;AK6;AK9;AK12;AK15	0.79	PID459	AK2;AK3;AK10;AK13;AK15	0.49	PID283	AK2;AK3	0.20
PID615	AK2;AK5;AK6;AK7;AK9;AK10;AK11;AK13	0.79	PID462	AK2;AK3;AK4;AK6;AK13	0.49	PID296	AK2;AK14	0.20
PID679	AK2;AK5;AK7;AK9;AK11;AK12;AK13;AK14	0.79	PID463	AK2;AK3;AK4;AK5;AK11	0.49	PID317	AK2;AK11	0.20
PID686	AK2;AK3;AK6;AK7;AK8;AK14;AK15;AK18	0.79	PID468	AK2;AK3;AK4;AK5;AK8	0.49	PID318	AK3;AK5	0.20
PID687	AK2;AK3;AK5;AK6;AK7;AK8;AK16;AK18	0.79	PID481	AK2;AK3;AK6;AK7;AK12	0.49	PID319	AK3;AK14	0.20
PID	AK2;AK3;AK6;	0.79	PID4	AK2;AK3;A	0.49	PID3	AK2;AK3	0.20

691	AK7;AK8;AK11;AK12;AK14		89	K6;AK7;AK14		20		
PID692	AK2;AK3;AK4;AK5;AK7;AK8;AK10;AK17	0.79	PID490	AK2;AK3;AK7;AK8;AK12	0.49	PID333	AK8;AK14	0.20
PID698	AK3;AK4;AK5;AK9;AK10;AK11;AK13;AK14	0.79	PID492	AK2;AK3;AK5;AK7;AK11	0.49	PID334	AK3;AK14	0.20
PID708	AK2;AK3;AK4;AK5;AK7;AK11;AK17;AK18	0.79	PID493	AK2;AK3;AK6;AK7;AK11	0.49	PID342	AK2;AK4	0.20
PID713	AK2;AK3;AK4;AK6;AK8;AK9;AK12;AK21	0.79	PID496	AK2;AK6;AK9;AK12;AK13	0.49	PID350	AK2;AK10	0.20
PID717	AK2;AK3;AK5;AK6;AK8;AK10;AK15;AK17	0.79	PID498	AK2;AK3;AK7;AK12;AK16	0.49	PID353	AK2;AK5	0.20
PID720	AK2;AK6;AK8;AK9;AK10;AK12;AK13;AK15	0.79	PID508	AK2;AK3;AK4;AK5;AK18	0.49	PID361	AK2;AK18	0.20
PID732	AK3;AK4;AK6;AK9;AK10;AK13;AK14;AK19	0.79	PID512	AK2;AK4;AK5;AK7;AK15	0.49	PID420	AK3;AK4	0.20
PID744	AK2;AK3;AK4;AK7;AK9;AK12;AK13;AK16	0.79	PID515	AK2;AK5;AK10;AK16;AK18	0.49	PID429	AK3;AK14	0.20
PID99	AK2;AK3;AK6;AK8;AK9;AK13;AK19	0.69	PID519	AK2;AK3;AK4;AK14;AK15	0.49	PID451	AK3;AK14	0.20
PID108	AK2;AK3;AK5;AK6;AK7;AK12;AK13	0.69	PID520	AK2;AK3;AK5;AK8;AK12	0.49	PID474	AK2;AK6	0.20
PID114	AK2;AK3;AK4;AK7;AK10;AK11;AK17	0.69	PID527	AK2;AK3;AK6;AK10;AK12	0.49	PID475	AK2;AK8	0.20
PID121	AK2;AK3;AK4;AK7;AK12;AK17;AK19	0.69	PID528	AK2;AK3;AK4;AK8;AK9	0.49	PID480	AK2;AK18	0.20
PID125	AK2;AK4;AK6;AK7;AK9;AK10;AK17	0.69	PID529	AK2;AK3;AK4;AK5;AK15	0.49	PID484	AK2;AK3	0.20
PID129	AK2;AK3;AK4;AK5;AK6;AK7;AK8	0.69	PID530	AK2;AK3;AK4;AK5;AK15	0.49	PID487	AK2;AK4	0.20
PID131	AK2;AK3;AK5;AK7;AK8;AK16;AK17	0.69	PID531	AK2;AK3;AK7;AK13;AK15	0.49	PID523	AK2;AK3	0.20
PID133	AK2;AK3;AK4;AK6;AK12;AK15;AK17	0.69	PID536	AK2;AK7;AK8;AK9;AK15	0.49	PID574	AK2;AK7	0.20
PID172	AK2;AK3;AK4;AK8;AK9;AK13;AK20	0.69	PID570	AK2;AK3;AK6;AK7;AK11	0.49	PID614	AK4;AK10	0.20
PID	AK2;AK3;AK4;	0.69	PID5	AK2;AK4;A	0.49	PID6	AK2;AK11	0.20

174	AK5;AK6;AK9;AK22		71	K6;AK9;AK15		20		
PID187	AK2;AK3;AK6;AK9;AK13;AK14;AK15	0.69	PID587	AK2;AK3;AK4;AK10;AK20	0.49	PID630	AK2;AK14	0.20
PID188	AK2;AK3;AK4;AK11;AK12;AK13;AK19	0.69	PID603	AK2;AK3;AK5;AK8;AK12	0.49	PID633	AK14;AK21	0.20
PID199	AK2;AK3;AK4;AK6;AK9;AK11;AK12	0.69	PID623	AK2;AK3;AK5;AK9;AK16	0.49	PID634	AK2;AK14	0.20
PID212	AK2;AK3;AK5;AK9;AK10;AK13;AK14	0.69	PID624	AK2;AK3;AK4;AK7;AK10	0.49	PID635	AK2;AK20	0.20
PID215	AK2;AK4;AK5;AK7;AK8;AK10;AK17	0.69	PID641	AK2;AK3;AK4;AK5;AK14	0.49	PID703	AK13;AK21	0.20
PID223	AK2;AK3;AK4;AK5;AK8;AK9;AK16	0.69	PID650	AK2;AK3;AK7;AK8;AK17	0.49	PID739	AK2;AK5	0.20
PID226	AK2;AK3;AK4;AK5;AK6;AK11;AK13	0.69	PID656	AK2;AK7;AK10;AK17;AK20	0.49	PID751	AK2;AK8	0.20
PID230	AK2;AK3;AK7;AK9;AK10;AK13;AK15	0.69	PID658	AK2;AK3;AK4;AK11;AK12	0.49	PID82	AK2	0.10
PID235	AK2;AK3;AK4;AK5;AK8;AK9;AK13	0.69	PID669	AK2;AK4;AK6;AK9;AK15	0.49	PID83	AK2	0.10
PID248	AK2;AK3;AK4;AK7;AK9;AK10;AK13	0.69	PID671	AK2;AK3;AK8;AK12;AK15	0.49	PID98	AK6	0.10
PID257	AK2;AK3;AK4;AK6;AK14;AK16;AK21	0.69	PID676	AK2;AK3;AK6;AK11;AK15	0.49	PID163	AK2	0.10
PID258	AK2;AK4;AK6;AK7;AK8;AK9;AK21	0.69	PID680	AK2;AK4;AK8;AK9;AK21	0.49	PID165	AK14	0.10
PID273	AK2;AK3;AK7;AK9;AK10;AK15;AK17	0.69	PID688	AK2;AK4;AK8;AK10;AK15	0.49	PID181	AK2	0.10
PID274	AK2;AK4;AK5;AK6;AK7;AK10;AK17	0.69	PID702	AK2;AK3;AK11;AK14;AK16	0.49	PID201	AK2	0.10
PID293	AK2;AK3;AK5;AK10;AK11;AK12;AK15	0.69	PID712	AK2;AK3;AK4;AK12;AK17	0.49	PID241	AK5	0.10
PID306	AK2;AK3;AK5;AK7;AK9;AK10;AK17	0.69	PID714	AK2;AK3;AK4;AK5;AK9	0.49	PID302	AK2	0.10
PID315	AK2;AK3;AK4;AK5;AK9;AK10;AK14	0.69	PID721	AK2;AK4;AK5;AK10;AK22	0.49	PID304	AK2	0.10
PID	AK2;AK3;AK5;	0.69	PID7	AK2;AK3;A	0.49	PID3	AK14	0.10

341	AK8;AK12;AK13;AK15		28	K4;AK10;AK21		29		
PID362	AK2;AK3;AK5;AK6;AK7;AK8;AK17	0.69	PID731	AK2;AK3;AK4;AK5;AK8	0.49	PID336	AK8	0.10
PID365	AK2;AK3;AK4;AK10;AK11;AK13;AK19	0.69	PID736	AK2;AK3;AK7;AK10;AK14	0.49	PID338	AK14	0.10
PID368	AK2;AK3;AK4;AK5;AK6;AK7;AK11	0.69	PID746	AK2;AK3;AK5;AK9;AK14	0.49	PID371	AK14	0.10
PID374	AK2;AK3;AK5;AK7;AK9;AK12;AK15	0.69	PID749	AK2;AK3;AK4;AK12;AK21	0.49	PID372	AK5	0.10
PID380	AK2;AK3;AK4;AK7;AK8;AK15;AK16	0.69	PID77	AK2;AK6;AK7;AK8	0.39	PID428	AK2	0.10
PID383	AK2;AK3;AK9;AK10;AK11;AK12;AK13	0.69	PID78	AK2;AK5;AK6;AK10	0.39	PID433	AK2	0.10
PID387	AK2;AK3;AK4;AK7;AK12;AK13;AK15	0.69	PID91	AK2;AK3;AK11;AK14	0.39	PID438	AK3	0.10
PID395	AK2;AK4;AK7;AK12;AK16;AK19;AK21	0.69	PID96	AK2;AK8;AK10;AK14	0.39	PID452	AK3	0.10
PID398	AK2;AK3;AK8;AK9;AK12;AK15;AK16	0.69	PID109	AK2;AK4;AK8;AK10	0.39	PID467	AK2	0.10
PID412	AK2;AK4;AK6;AK7;AK11;AK13;AK19	0.69	PID115	AK2;AK3;AK4;AK15	0.39	PID471	AK2	0.10
PID415	AK2;AK3;AK6;AK9;AK12;AK15;AK17	0.69	PID117	AK2;AK4;AK5;AK19	0.39	PID482	AK2	0.10
PID416	AK3;AK4;AK7;AK12;AK15;AK16;AK19	0.69	PID127	AK2;AK4;AK5;AK18	0.39	PID485	AK2	0.10
PID434	AK2;AK4;AK5;AK6;AK8;AK12;AK13	0.69	PID128	AK2;AK3;AK7;AK11	0.39	PID585	AK2	0.10
PID435	AK3;AK4;AK5;AK9;AK13;AK16;AK18	0.69	PID139	AK2;AK10;AK14;AK18	0.39	PID632	AK2	0.10
PID449	AK2;AK3;AK4;AK5;AK8;AK14;AK16	0.69	PID141	AK2;AK3;AK7;AK16	0.39	PID644	AK2	0.10
PID469	AK2;AK3;AK4;AK6;AK8;AK11;AK13	0.69	PID142	AK2;AK4;AK12;AK13	0.39	PID689	AK2	0.10
PID473	AK2;AK3;AK7;AK8;AK12;AK14;AK16	0.69	PID144	AK2;AK3;AK9;AK18	0.39	PID727	AK2	0.10
PID	AK2;AK3;AK4;	0.69	PID1	AK2;AK4;A	0.39	PID7	AK2	0.10

501	AK5;AK6;AK11;AK15		47	K11;AK12		50		
PID504	AK2;AK3;AK5;AK9;AK11;AK14;AK16	0.69	PID150	AK2;AK4;AK5;AK9	0.39	PID343		0.00
PID509	AK2;AK4;AK5;AK8;AK10;AK12;AK15	0.69	PID151	AK2;AK3;AK7;AK9	0.39	PID345		0.00
PID511	AK2;AK3;AK5;AK6;AK7;AK11;AK14	0.69	PID152	AK2;AK3;AK4;AK17	0.39	PID629		0.00
PID526	AK2;AK3;AK5;AK8;AK10;AK12;AK15	0.69	PID156	AK2;AK3;AK7;AK13	0.39			

CHAPTER 4

Hamming Code Generator/Checker using QCA

Ayan Chaudhuri, Diganta Sengupta, Chitrita Chaudhuri, "Adiabatic Hamming Code Generator-Checker using 4-Dot-2-Electron Quantum Dot Cellular Automata", International Journal of Engineering and Advanced Technology, vol. 8, issue 6, pp 4435-4439, August 2019. (SCOPUS Indexed).

4 HAMMING CODE GENERATOR/CHECKER USING QCA

QCA [**50**] [**62**] [**3**] [**51**] has gained momentum in the last couple of decades due to the fact that it exhibits partial reversibility. Moreover the feature of near zero heat dissipation has contributed to the popularity of QCA. A lot of Boolean specifications have been designed using QCA. It has been possible because the basic Boolean gates, viz. NOT, AND, and OR can be experimentally realized using the QCA cells [**63**]. Since, the basic Boolean gates can be implemented using QCA, hence any digital circuit can be designed using QCA cells. Apart from the fundamental Boolean gates, the XOR gate [**64**] [**65**] [**66**] forms another important gate due to its huge alliance to cryptography. In this chapter, the XOR gate is of interest since the XOR gates form the fundamental requirement for designing Hamming Code converters. A few proposals for Hamming Code converter exists in literature [**67**] [**68**] [**69**], but it can be observed that the proposal in this chapter fares better than all of them, as will be seen in the analysis section.

As the XOR gate is inevitable in Hamming Code designs, hence three literary proposals of XOR gate are presented in Figure 4.1, Figure 4.2, and Figure 4.3. The XOR gate proposal in Figure 4.3 has been utilized in this chapter to design the Hamming Code Generator/Checker circuit.

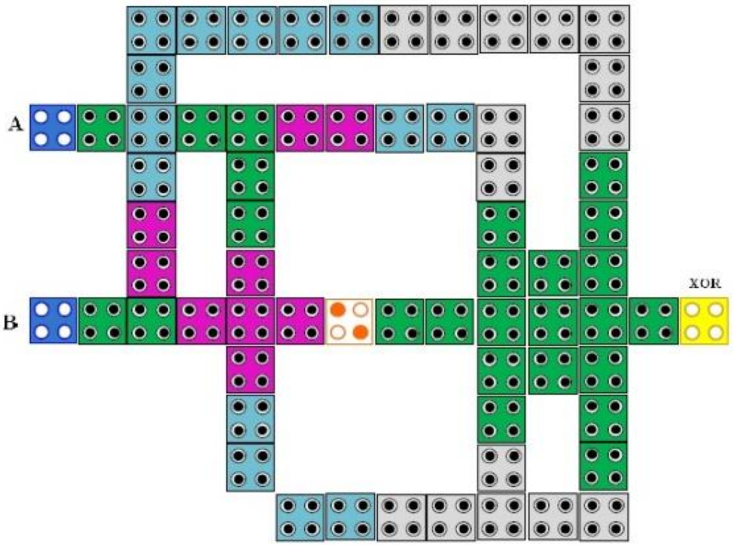


Figure 4.1. XOR Gate design of [**64**]

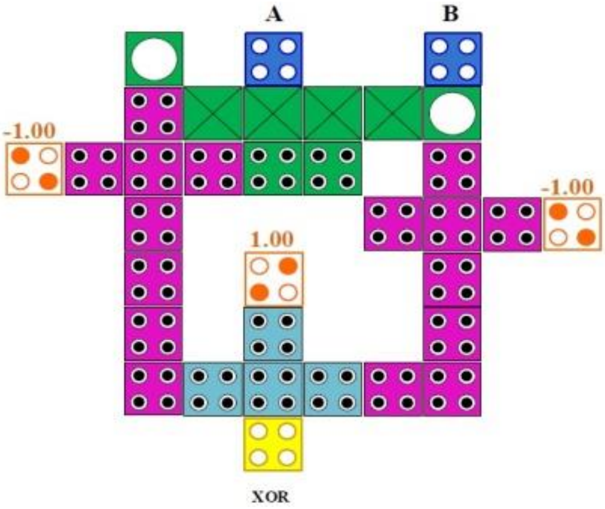


Figure 4.2. XOR Gate design of [**65**]

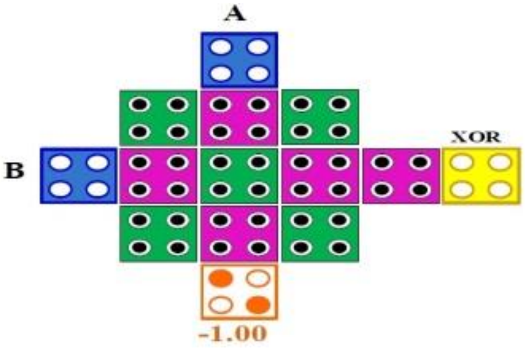


Figure 4.3. XOR Gate design of [**66**]

It can be observed from Figure 4.1, Figure 4.2, and Figure 4.3 that the latter consumes the least amount of QCA cells. Also as will be discussed later, the design in Figure 4.3 has the least latency among all the three XOR implementations. Also since the design in Figure 4.2 implements wire crossovers, this chapter concentrates on coplanar wires, hence the design is discarded. Hamming code converter based on NAND/NOR implementations have been proposed in [**67**]. The Hamming Code Generator/Checker proposed in [**67**] uses the NAND and the NOR gates. The authors have grouped the design into groups of $3 \times 3, 3 \times 2$ and 2×2 zone, and routed the QCA cells through them. Another proposal that uses the same technology presented in [**67**] is proposed in [**69**].

4.1 THE PROPOSED HAMMING CODE CONVERTER

Hamming code is generally used for single error correction and is known as Forward Error Correcting codes. The traditional process of error detection as well as error correction logic using Hamming Code is presented below in Equation (1) - Equation (12).

Let

$$Message\ P = p_7p_6p_5p_4p_3p_2p_1p_0 \quad (1)$$

$$Coded\ Message\ C = p_7p_6p_5p_4h_8p_3p_2p_1h_4p_0h_2h_1 \quad (2)$$

Where P denotes the Hamming Code bits $h_1, h_2, h_4, and\ h_8$.

Let, X denoted the Hamming Coded message C. Hence,

$$X = \sum_{i=0}^{11} x_{i+1}2^i$$

$$\Rightarrow D = x_{12}x_{11}x_{10}x_9x_8x_7x_6x_5x_4x_3x_2x_1 \quad (3)$$

Where

$x_1 = h_1$; $x_2 = h_2$; $x_3 = p_0$; $x_4 = h_4$; $x_5 = p_1$; $x_6 = p_2$

$x_7 = p_3$; $x_8 = h_8$; $x_9 = p_4$; $x_{10} = p_5$; $x_{11} = p_6$; $x_{12} = p_7$

The Hamming code bits are generated according to the given equations - Equation (4) to Equation (7).

$$h_1 = x_1 \oplus x_3 \oplus x_5 \oplus x_7 \oplus x_9 \oplus x_{11} \quad (4)$$

$$h_2 = x_2 \oplus x_3 \oplus x_6 \oplus x_7 \oplus x_{10} \oplus x_{11} \quad (5)$$

$$h_4 = x_4 \oplus x_5 \oplus x_6 \oplus x_7 \oplus x_{12} \quad (6)$$

$$h_8 = x_8 \oplus x_9 \oplus x_{10} \oplus x_{11} \oplus x_{12} \quad (7)$$

Where, the binary XOR operation is denoted by ⊕.

Equation (8) – Equation (11) generate the Hamming Code Checker bits.

$$c_1 = x_1 \oplus x_3 \oplus x_5 \oplus x_7 \oplus x_9 \oplus x_{11} \oplus h_1 \quad (8)$$

$$c_2 = x_2 \oplus x_3 \oplus x_6 \oplus x_7 \oplus x_{10} \oplus x_{11} \oplus h_2 \quad (9)$$

$$c_4 = x_4 \oplus x_5 \oplus x_6 \oplus x_7 \oplus x_{12} \oplus h_4 \quad (10)$$

$$c_8 = x_8 \oplus x_9 \oplus x_{10} \oplus x_{11} \oplus x_{12} \oplus h_8 \quad (11)$$

Let

$H = h_8 h_4 h_2 h_1 \; and \; C' = c_8 c_4 c_2 c_1$

Hence, if there is any error in the transmission, then the error message is generated as follows:

$$E = \sum_{i=0}^{3} e_i 2^i \;\; ; where \; e_i = h_j \oplus c_j \forall j \in \{8,4,2,1\} \;\; (12)$$

The Error message E points out the error at the bit position in the transmitted message D. The whole process is illustrated in Illustration 1.

4.1.1 ILLUSTRATION 1

Let $M = 10100111$ be the initial message signal $P = p_7 p_6 p_5 p_4 p_3 p_2 p_1 p_0$.

Hence,

$C = 1010h_8011h_41h_2h_1$ and $D = 1010h_8011h_41h_2h_1$ after equating Equation (4) to Equation (7).

Therefore $h_1 = 0$, $h_2 = 1$, $h_4 = 1$, and $h_8 = 0$ respectively.

Hence, the transmitted message $D = 101000111110$

Now, let us assume that the bit at $x_6\ of\ D$ gets corrupted.

Therefore the received signal is $D = 101000\mathbf{0}11110$. The corrupted bit is represented by the bold and blue colored bit.

At the receiver end, Equation (8) - Equation (11) generate the Checker bits.

Hence,

$c_1 = 0, c_2 = 0, c_4 = 0, and\ c_8 = 0$ respectively.

Now, the error is calculated according to Equation (12).

Hence, $E = 0110$, where

$e_4 = h_8 \oplus c_8 = 0$

$e_3 = h_4 \oplus c_4 = 1$

$e_2 = h_2 \oplus c_2 = 1$

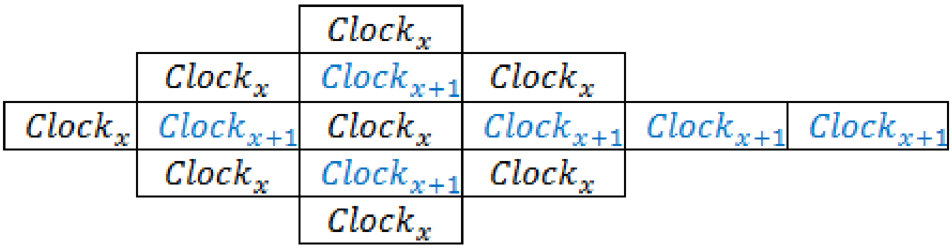
$e_1 = h_1 \oplus c_1 = 0$

Therefore, it can be said that the bit position 6 from the right in $D = x_{12}x_{11}x_{10}x_9x_8x_7x_6x_5x_4x_3x_2x_1$ has got corrupted during transmission.

Equations (4) through Equation (12) reflect that all the operations related to Hamming code bit generation and Hamming Check bit generation is implemented using XOR operations. Hence, the choice of the XOR gate presented in Figure 4.3 has been used to design the Hamming Code Generator/Checker in this chapter.

4.2 QCA CELL DESIGNS FOR HAMMING CODE GENERATOR/CHECKER

As discussed, the XOR gate of [**66**] presented in Figure 4.3 has been used to design the proposed QCA architectures. Figure 4.4 provides the clocking pattern for the XOR gate.



		$Clock_x$			
	$Clock_x$	$Clock_{x+1}$	$Clock_x$		
$Clock_x$	$Clock_{x+1}$	$Clock_x$	$Clock_{x+1}$	$Clock_{x+1}$	$Clock_{x+1}$
	$Clock_x$	$Clock_{x+1}$	$Clock_x$		
		$Clock_x$			

Figure 4.4. Clocking sequence for XOR gate in [**66**].

Figure 4.4 reflects that a single clock zone is enough for a single XOR operation. Hence, it can be concluded that the XOR gate has a latency of 0.25.

The architectures for the Hamming Code Generator/Checker have been designed using Bistable Approximation as the simulation engine in QCA Designer [**9**]. The simulation parameters are as follows:

Layer properties:

Cell Width	: 18 nm
Cell Height	: 18 nm
Dot Diameter	: 5 nm

Simulation Engine Parameters

No. of samples	: 12800
Radius of effect	: 65 nm
Relative Permittivity	: 12.9
Clock High	: $9.8e^{-022}$
Clock Low	: $3.8e^{-023}$
Clock Amplitude Factor	: 2
Layer Separation	: 11.5
Max. Iterations/sample	: 100

The following subsections provide the QCA designs for Equation (4) through Equation (7) along with their clocking sequences.

4.2.1 EQUATION (4): QCA DESIGN

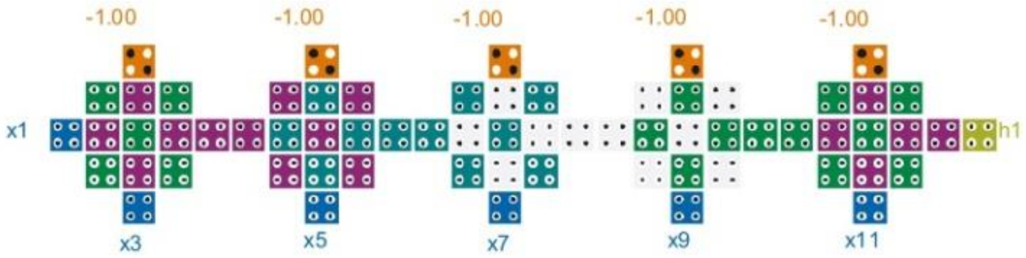
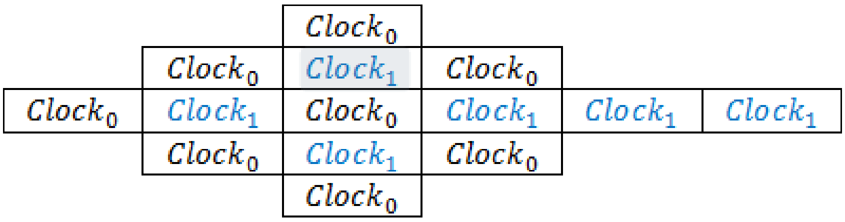
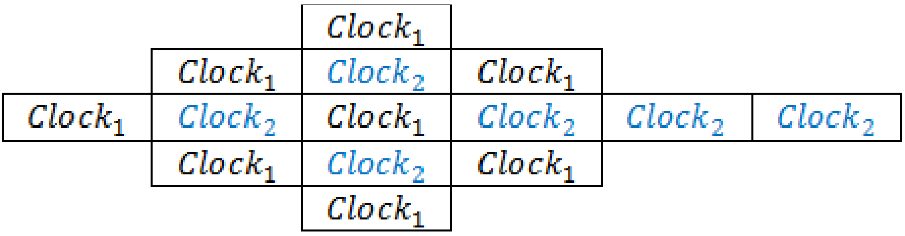


Figure 4.5. Equation (4): QCA Design



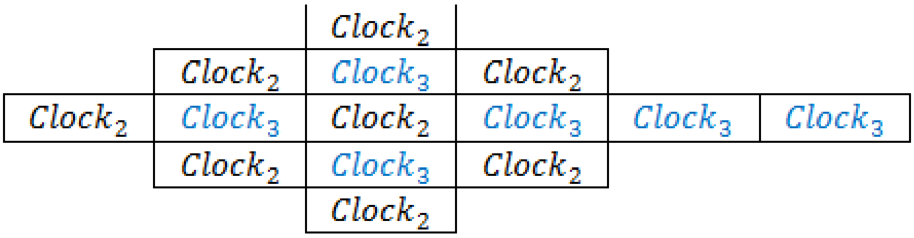
		$Clock_0$			
	$Clock_0$	$Clock_1$	$Clock_0$		
$Clock_0$	$Clock_1$	$Clock_0$	$Clock_1$	$Clock_1$	$Clock_1$
	$Clock_0$	$Clock_1$	$Clock_0$		
		$Clock_0$			

Figure 4.6. Clocking Sequence for first XOR gate in Figure 4.5



		$Clock_1$			
	$Clock_1$	$Clock_2$	$Clock_1$		
$Clock_1$	$Clock_2$	$Clock_1$	$Clock_2$	$Clock_2$	$Clock_2$
	$Clock_1$	$Clock_2$	$Clock_1$		
		$Clock_1$			

Figure 4.7. Clocking Sequence for second XOR gate in Figure 4.5



		$Clock_2$			
	$Clock_2$	$Clock_3$	$Clock_2$		
$Clock_2$	$Clock_3$	$Clock_2$	$Clock_3$	$Clock_3$	$Clock_3$
	$Clock_2$	$Clock_3$	$Clock_2$		
		$Clock_2$			

Figure 4.8. Clocking Sequence for third XOR gate in Figure 4.5

		$Clock_3$			
	$Clock_3$	$Clock_4$	$Clock_3$		
$Clock_3$	$Clock_4$	$Clock_3$	$Clock_4$	$Clock_4$	$Clock_4$
	$Clock_3$	$Clock_4$	$Clock_3$		
		$Clock_3$			

Figure 4.9. Clocking Sequence for fourth XOR gate in Figure 4.5

		$Clock_4$			
	$Clock_4$	$Clock_0$	$Clock_4$		
$Clock_4$	$Clock_0$	$Clock_4$	$Clock_0$	$Clock_0$	$Clock_0$
	$Clock_4$	$Clock_0$	$Clock_4$		
		$Clock_4$			

Figure 4.10. Clocking Sequence for fifth XOR gate in Figure 4.5

A total of five clock ones have been utilized in the design presented in Figure 4.6 – Figure 4.10 for generation of h_1. Hence, the latency for generation of h_1 is 1.25.

The next sub-sections provide the QCA implementations for the Hamming Code generator and checker bits.

4.2.2 EQUATION (5): QCA DESIGN

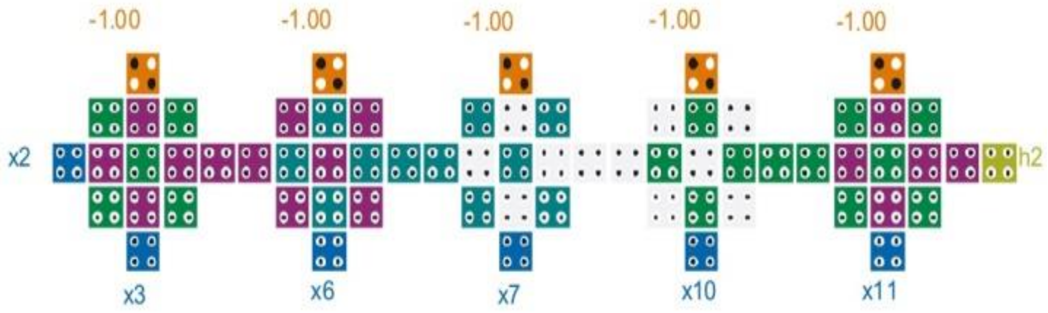


Figure 4.11. Architecture for generation of h_2 using QCA cells

4.2.3 EQUATION (6): QCA DESIGN

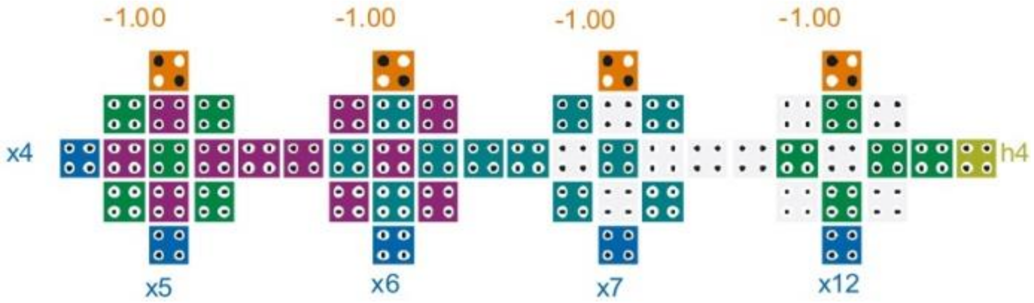


Figure 4.12. Architecture for generation of h_3 using QCA cells

4.2.4 EQUATION (7): QCA DESIGN

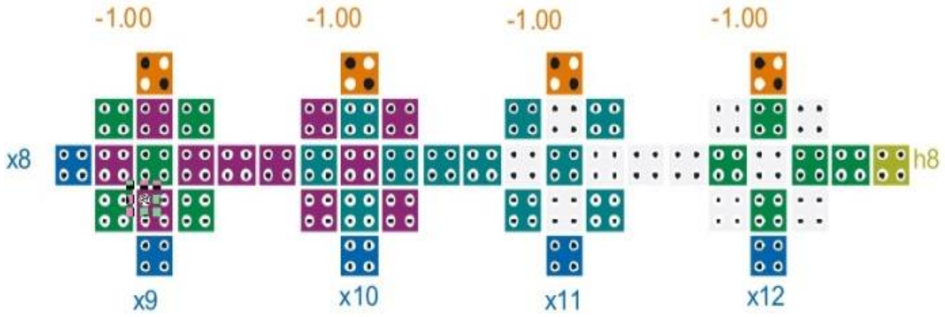


Figure 4.13. Architecture for generation of h_4 using QCA cells

4.2.5 EQUATION (8): QCA DESIGN

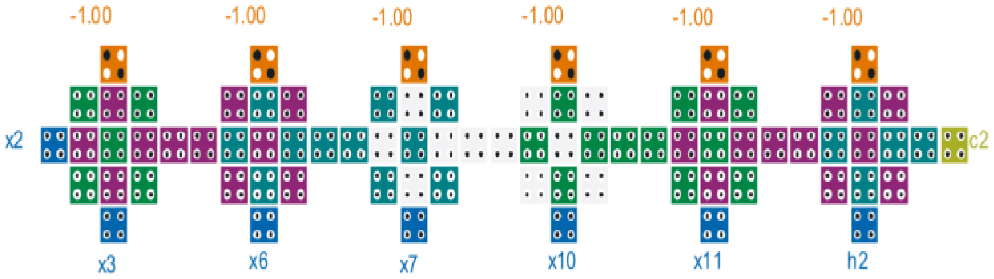


Figure 4.14. Architecture for generation of c_1 using QCA cells

4.2.6 EQUATION (9): QCA DESIGN

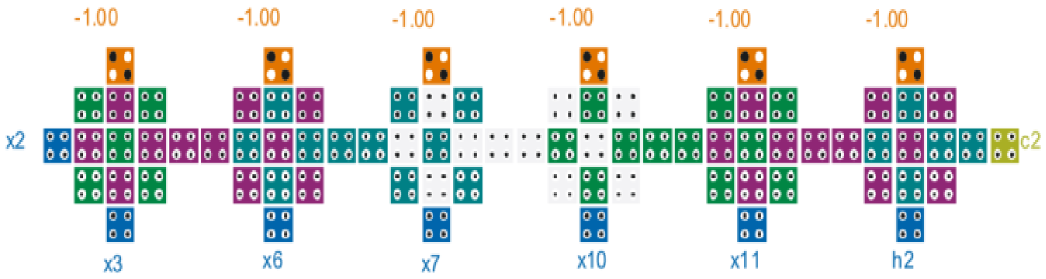


Figure 4.15. Architecture for generation of c_2 using QCA cells

4.2.7 EQUATION (10): QCA DESIGN

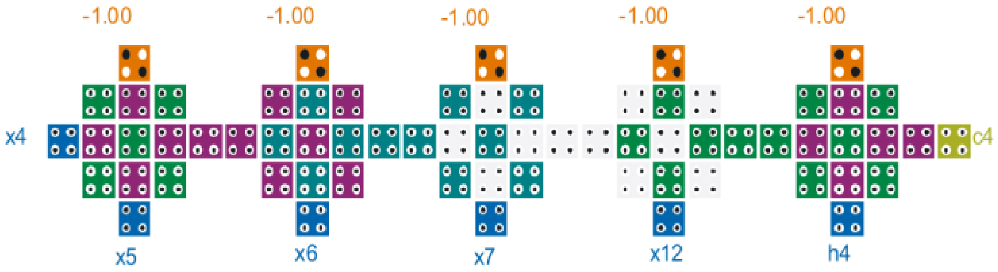


Figure 4.16. Architecture for generation of c_4 using QCA cells

4.2.8 EQUATION (11): QCA DESIGN

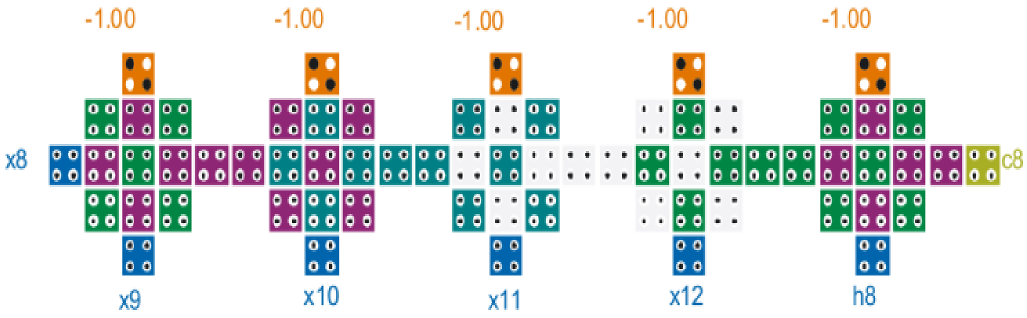


Figure 4.17. Architecture for generation of c_8 using QCA cells

4.2.9 EQUATION (12): QCA DESIGN

This subsection provides the complete architecture for generation of the error signal.

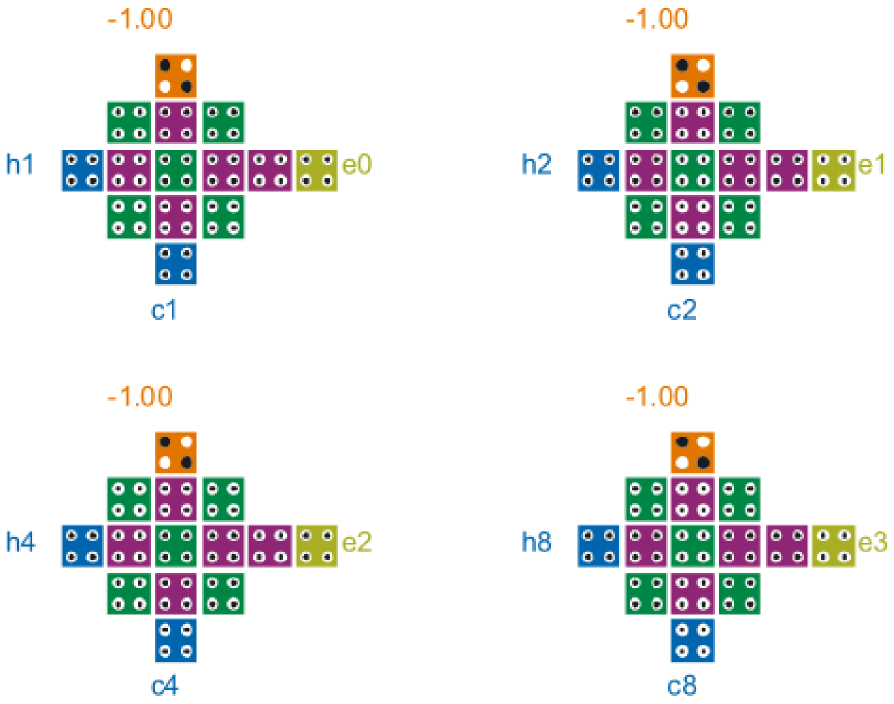
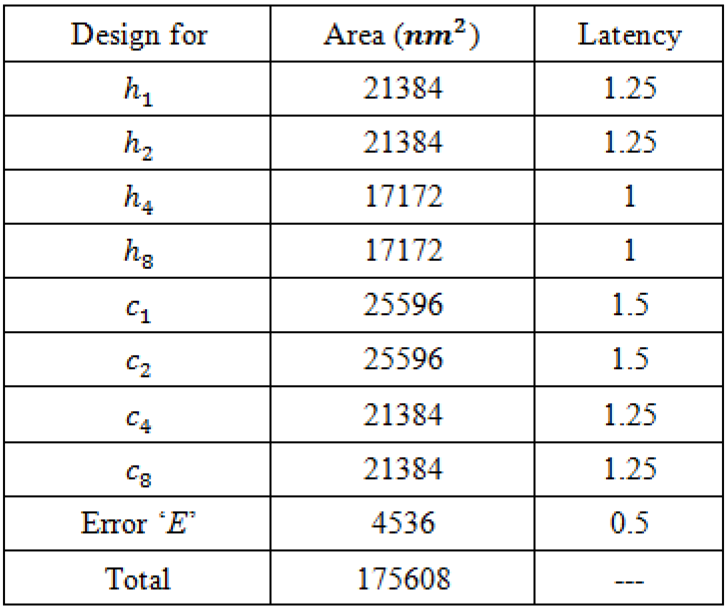


Figure 4.18. Architecture for generation of $E = \sum_{i=0}^{3} e_i 2^i$ using QCA cells

4.3 Design Analysis

The design analysis for the QCA architectures presented in this chapter is shown in Table 4.1.

TABLE 4.1. Design Analysis



Design for	Area (nm^2)	Latency
h_1	21384	1.25
h_2	21384	1.25
h_4	17172	1
h_8	17172	1
c_1	25596	1.5
c_2	25596	1.5
c_4	21384	1.25
c_8	21384	1.25
Error 'E'	4536	0.5
Total	175608	---

As discussed earlier, it has been seen that a single XOR gate has a latency of 0.25. Therefore Table 4.1 reflects that the highest latency is that of 1.25 in case of Hamming Code generator and 1.5 in case of Hamming Code Checker. Since, the operations of generation is parallel, hence the latency of the Hamming Code generator circuit is 1.25 whereas that of Hamming Code Checker circuit is 1.5.

The XOR gate proposal in [**69**] consumes a total of 33 cells with respect to the 13 cells required for the realization in [**66**], hence the proposed design in this chapter has an area gain of 60.6% over the proposal in [**69**]. Since, in [**69**], a single XOR gate consumes 1 latency, therefore the complete Hamming Code Generation operation will require 5 to 6 latencies. Thus the proposed generator in this chapter has a latency gain of 75% - 79.19% with respect to [**69**].

CHAPTER 5

Comprehensive analysis of CNF and XOR-AND

Ayan Chaudhuri, Mahamuda Sultana, Diganta Sengupta, Chitrita Chaudhuri and Atal Chaudhuri "A Comparative Analysis of CNF and XOR-AND Representations for QCA Majority Gate Estimation", *52nd Annual Convention of Computer Society of India (CSI 2017)*, January 19-21, 2018, Science City, Kolkata, Springer Nature CCIS Series, vol 836, pp 402-415. (SCOPUS)

5 COMPREHENSIVE ANALYSIS OF CNF AND XOR-AND

This chapter concentrates on the Majority Voter count in a QCA circuit. Basically the chapter targets to provide an analysis for the number of Majority Voters required to design a given Boolean specification using QCA cells, where the design has been minimized using K-Map approach and a non-conventional XOR-AND approach. The non-optimized approach for XOR-AND estimation has been proposed in Section 6.3 of [**70**] by Dajani. This chapter illustrates the algorithm provided by Dajani and proposed the optimized version for the algorithm. Dajani has proposed only a non-optimized version. The chapter also proposes a K-Map method for Majority Voter estimation and presents a comparative analysis for the Majority Voter count in all the designs.

5.1 BACKGROUND - NON-OPTIMIZED ALGORITHM FOR XOR-AND EXTRACTION

This section discusses the algorithm provided in Section 6.3 of [**70**] for extraction of XOR-AND combination gates from any given Boolean function.

Let the Boolean function be

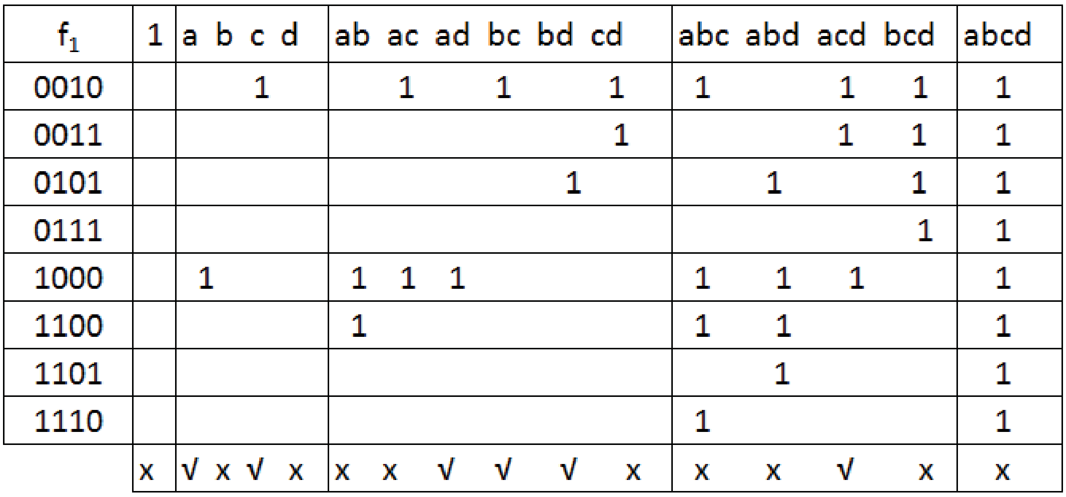
$$f = \sum (2,3,5,7,8,12,13,14) \quad (1)$$

The process of XOR-AND extraction is presented by the following algorithm step wise.

- The first step is to draw the minimization chart which contains all the minterms of the function f in the column named f_1 in Table 5.1.
- In the second step, the minterms are assumed to be having four bits $abcd$. In the first row of Table 5.1, all the possible values of the minterms are ticked where the ‘1’ in the first minterm exists. So, starting with 1, all the possible combinations are ticked where the relevant bit (out of $abcd$) can be ‘1’.
- The third step extracts the XOR-AND. This is done by putting a ‘x’ mark in all the columns which have even number of 1’s. It is to be noted that zero is

considered to have even number of 1’s.

TABLE 5.1. MINIMIZATION CHART (COURTESY [70])



f_1	1	a	b	c	d	ab	ac	ad	bc	bd	cd	abc	abd	acd	bcd	abcd
0010				1			1		1		1	1		1	1	1
0011											1			1	1	1
0101										1			1		1	1
0111															1	1
1000		1				1	1	1				1	1	1		1
1100						1						1	1			1
1101													1			1
1110												1				1
	x	√	x	√	x	x	x	√	√	√	x	x	x	√	x	x

Hence using Table 5.1, for the Boolean function

$$f = \sum(2,3,5,7,8,12,13,14) \quad (1)$$

The XOR-AND expression comes to

$$f = a \oplus c \oplus ad \oplus bc \oplus bd \oplus acd \quad (2)$$

5.2 OPTIMIZATION OF XOR-AND EXPRESSION

After retrieving the XOR-AND expression as discussed in the previous section, a set of optimization rules have been generated using the XOR function truth table. The optimization rule is presented in Table 5.2. When the substitutions provided in Table 5.2 are used, a minimized XOR-AND expression is obtained.

TABLE 5.2. XOR-AND OPTIMIZATION RULES

Expression	Optimized Expression	Expression	Optimized Expression	Expression	Optimized Expression
$abcd \oplus abc$	$abcd'$	$bcd \oplus b$	$(cd)'b$	$abc \oplus bc$	$a'bc$
$abcd \oplus abd$	$abc'd$	$bcd \oplus c$	$(bd)'c$	$abc \oplus a$	$(bc)'a$
$abcd \oplus acd$	$ab'cd$	$bcd \oplus d$	$(bc)'d$	$abc \oplus b$	$(ac)'b$
$abcd \oplus bcd$	$a'bcd$	$acd \oplus ac$	acd'	$abc \oplus c$	$(ab)'c$
$abcd \oplus ab$	$(cd)'ab$	$acd \oplus ad$	$ac'd$	$cd \oplus c$	cd'
$abcd \oplus ac$	$(bd)'ac$	$acd \oplus cd$	$a'cd$	$cd \oplus d$	$c'd$
$abcd \oplus ad$	$(bc)'ad$	$acd \oplus a$	$(cd)'a$	$bd \oplus b$	bd'
$abcd \oplus bc$	$(ad)'bc$	$acd \oplus c$	$(ad)'c$	$bd \oplus d$	$b'd$
$abcd \oplus bd$	$(ac)'bd$	$acd \oplus d$	$(ac)'d$	$bc \oplus b$	bc'
$abcd \oplus cd$	$(ab)'cd$	$abd \oplus ab$	abd'	$bc \oplus c$	$b'c$
$abcd \oplus a$	$(bcd)'a$	$abd \oplus ad$	$ab'd$	$ad \oplus a$	ad'
$abcd \oplus b$	$(acd)'b$	$abd \oplus bd$	$a'bd$	$ad \oplus d$	$a'd$
$abcd \oplus c$	$(abd)'c$	$abd \oplus a$	$(bd)'a$	$ac \oplus a$	ac'
$abcd \oplus d$	$(abc)'d$	$abd \oplus b$	$(ad)'b$	$ac \oplus c$	$a'c$
$bcd \oplus bc$	bcd'	$abd \oplus d$	$(ab)'d$	$ab \oplus a$	ab'
$bcd \oplus bd$	$bc'd$	$abc \oplus ab$	abc'	$ab \oplus b$	$a'b$
$bcd \oplus cd$	$b'cd$	$abc \oplus ac$	$ab'c$		

It can be observed from Table 5.2 that the optimization of XOR-AND expressions is being converted into a AND-NOT expression. The AND expressions have been used to calculate the number of majority voters as per the rule provided in Table 5.3. The rules presented in Table 5.3 are for four variable functions but can be extended for any number of variables.

TABLE 5.3. RULES FOR MAJORITY VOTER REALIZATIONS OF AND EXPRESSIONS

AND Expression	Majority Voter expression	MV count
$abcd$	$M_1\ (a, b, 0) \rightarrow M_2\ (ab, c, 0) \rightarrow M_3\ (abc, d, 0)$	3
abc	$M_1\ (a, b, 0) \rightarrow M_2\ (ab, c, 0)$	2
abd	$M_1\ (a, b, 0) \rightarrow M_2\ (ab, d, 0)$	2
acd	$M_1\ (a, c, 0) \rightarrow M_2\ (ac, d, 0)$	2
bcd	$M_1\ (b, c, 0) \rightarrow M_2\ (bc, d, 0)$	2
ab	$M_1\ (a, b, 0)$	1
ac	$M_1\ (a, c, 0)$	1
ad	$M_1\ (a, d, 0)$	1
bc	$M_1\ (b, c, 0)$	1
bd	$M_1\ (b, d, 0)$	1
cd	$M_1\ (c, d, 0)$	1

Where

$$M_i(\mathrm{x,y,z}) = \mathrm{xy + xz + yz} \quad (3)$$

The Boolean OR function is presented by the $'+'$ symbol, and the input to the M_{i+1} majority voter gate is the output of M_i, the i^{th}majority voter gate.

It can be observed from Table 5.3 that three and two majority voters are required for a four and three variable minterms respectively.

Hence Table 5.3 also provides the number of majority voter for a given Boolean function.

The majority voter count for all the minterms in Table 5.3 are provided in Table 5.4.

TABLE 5.4. NUMBER OF MAJORITY VOTER WITH RESPECT TO MINTERMS

Boolean expression	Majority Voter Count	Boolean expression	Majority Voter Count	Boolean expression	Majority Voter Count
$abcd'$	3	$(cd)'b$	2	$a'bc$	2
$abc'd$	3	$(bd)'c$	2	$(bc)'a$	2
$ab'cd$	3	$(bc)'d$	2	$(ac)'b$	2
$a'bcd$	3	acd'	2	$(ab)'c$	2
$(cd)'ab$	3	$ac'd$	2	cd'	1
$(bd)'ac$	3	$a'cd$	2	$c'd$	1
$(bc)'ad$	3	$(cd)'a$	2	bd'	1
$(ad)'bc$	3	$(ad)'c$	2	$b'd$	1
$(ac)'bd$	3	$(ac)'d$	2	bc'	1
$(ab)'cd$	3	abd'	2	$b'c$	1
$(bcd)'a$	3	$ab'd$	2	ad'	1
$(acd)'b$	3	$a'bd$	2	$a'd$	1
$(abd)'c$	3	$(bd)'a$	2	ac'	1
$(abc)'d$	3	$(ad)'b$	2	$a'c$	1
bcd'	2	$(ab)'d$	2	ab'	1
$bc'd$	2	abc'	2	$a'b$	1
$b'cd$	2	$ab'c$	2		

The following expression provides the majority voter count based on the minterm count in a simplified XOR-AND expression.

$$MV\ Count = \lfloor {}^{n}/_{2} \rfloor * 3 \quad (4)$$

where

$n = number\ of\ terms\ in\ the\ simplified\ XOR - AND\ Expression$

Therefore

$$MV_{Count} = MV_{Count-1} + MV_{Count-2} \quad (5)$$

Where

$MV_{Count-1}$ is provided by Table 5.4 and

$MV_{Count-2}$ is given by Equation (4).

The total majority voter count is given by Equation (5).

5.2.1 ILLUSTRATIVE EXAMPLE 1

Let the Boolean function be

$$f = \sum(2,3,5,7,8,12,13,14) \quad (6)$$

Equation (7) provides the XOR-AND expression for Equation (6) using Table 5.1 and subsection 5.2.

$$f = a \oplus c \oplus ad \oplus bc \oplus bd \oplus acd \quad (7)$$

Therefore,

$MV_{Count-1}\ for\ Equation\ (7) = 0 + 0 + 1 + 1 + 1 + 2 = 5\ (Table\ 5.4)$

$MV_{Count-2}\ for\ Equation\ (7) = 9\ using\ Equation\ (4)$

Therefore, $MV_{Count} = MV_{Count-1} + MV_{Count-2} = 5 + 9 = 14$

5.2.2 ILLUSTRATIVE EXAMPLE 2

Let the Boolean function be

$$f = a \oplus b \oplus c \oplus d \oplus ab \oplus \mathrm{ac} \oplus ad \oplus bc \oplus bd \oplus cd \oplus abc \oplus abd \oplus acd \oplus bcd \oplus abcd \quad (8)$$

Equation (9) provides the XOR-AND expression for Equation (6) using Table 5.1 and subsection 5.2.

$$f = a \oplus \mathrm{a}'b \oplus \mathrm{a}'c \oplus \mathrm{a}'d \oplus a'bc \oplus a'bd \oplus a'cd \oplus a'bcd \quad (9)$$

Therefore,

$MV_{Count-1}\ for\ Equation\ (9) = 0 + 1 + 1 + 1 + 2 + 2 + 2 + 3 = 12\ (Table\ 5.4)$

$MV_{Count-2}\ for\ Equation\ (7) = 12\ using\ Equation\ (4)$

Therefore, $MV_{Count} = MV_{Count-1} + MV_{Count-2} = 12 + 12 = 24$

5.3 K-MAP METHOD

In this method, a Boolean function in CNF is first simplified using the K-Map and then the majority voter count is taken. For all the minterms, the majority voter count is presented in Table 5.5.

TABLE 5.5. NUMBER OF MAJORITY VOTERS FOR EACH MINTERM

Boolean expression	Majority Voter Count	Boolean expression	Majority Voter Count	Boolean expression	Majority Voter Count
$a'c'$	1	$a'b'c'$	2	$b'c'd'$	2
$a'd$	1	$a'b'd$	2	$b'c'd$	2
$a'c$	1	$a'b'c$	2	$b'cd$	2
bc'	1	$a'bc'$	2	$b'cd'$	2
bd	1	$a'bd$	2	$ab'd'$	2
bc	1	$a'bc$	2	$a'bd'$	2
ac'	1	abc'	2	abd'	2
ad	1	abd	2	$ab'd'$	2
ac	1	abc	2	$a'b'c'd'$	3
$b'c'$	1	$ab'c'$	2	$a'b'c'd$	3
$b'd$	1	$ab'd$	2	$a'b'cd$	3
$b'c$	1	$ab'c$	2	$a'b'cd'$	3
$a'd'$	1	$a'c'd'$	2	$a'bc'd'$	3
bd'	1	$a'c'd$	2	$a'bc'd$	3
ad'	1	$a'cd$	2	$a'bcd$	3
$a'b'$	1	$a'cd'$	2	$a'bcd'$	3
$a'b$	1	$bc'd'$	2	$abc'd'$	3
ab	1	$bc'd$	2	$abc'd$	3
ab'	1	bcd	2	$abcd$	3
$c'd'$	1	bcd'	2	$abcd'$	3
$c'd$	1	$ac'd'$	2	$ab'c'd'$	3
cd	1	$ac'd$	2	$ab'c'd$	3
cd'	1	acd	2	$ab'cd$	3
$b'd'$	1	acd'	2	$ab'cd'$	3

$MV_{Count-1}\ for\ K-Map\ is\ given\ in\ Table$ 5.5.

$MV_{Count-2}\ for\ K-Map\ is\ given\ in\ Equation\ (10).$

$$MV_{Count-2} = n - 1 \quad (10)$$

And as usual

$$MV_{Count} = MV_{Count-1} + MV_{Count-2} \quad (5)$$

5.3.1 ILLUSTRATIVE EXAMPLE 3

Let the Boolean function be

$$f = \sum (2,3,5,7,8,12,13,14) \quad (11)$$

Equation (12) presents the simplified expression by K-Map

$$f = ac' + ab' + bcd' \quad (12)$$

$MV_{Count-1} for\ Equation\ (11)\ usingTable\ 5.5 = 1 + 1 + 2 = 4$

$MV_{Count-2}\ for\ Equation\ (12) = 2; using\ Equation\ (10)$

Therefore,

$MV_{Count} = MV_{Count-1} + MV_{Count-2} = 4 + 2 = 6.$

5.4 COMPARATIVE ANALYSIS

For the sake of comparative analysis, ten standard functions have been used. These ten functions have been implemented using both the algorithms, XOR-AND, and K-Map and the results are reflected in Table 5.6

TABLE 5.6. TEN STANDARD FUNCTIONS – COMPARATIVE ANALYSIS

SF	Number of Majority Voters - 1	Number of Majority Voters - 2
$f = abc$	2	2
$f = abc + a'b'c'$	12	5
$f = abc + ab'c'$	5	5
$f = ab + bc$	7	3
$f = ab + a'b'c$	11	4
$f = a'bc + a'b'c' + abc'$	9	8
$f = ab + bc + ac$	6	5
$f = ab + b'c$	5	3
$f = ab + bc + a'b'c'$	7	6
$f = abc + a'b'c + ab'b' + a'bc'$	3	11

SF: Standard Functions

Number of Majority Voters – 1 – MV Count using Optimized XOR-AND

Number of Majority Voters – 2 – MV Count using Karnaugh Map

The graphical representation of the comparative analysis is present in Figure 5.1. It can be observed from Figure 5.1 that there are different counts for majority voters in the two approaches discussed in both the algorithms. They do not follow any standard rule. Hence it is advised to follow both the rules and check out which algorithm provides the better results in terms of majority voter count.

It can be observed that in the worst case, for any given Boolean function in CNF, if the simplified equation using K-Map possesses eight minterms.

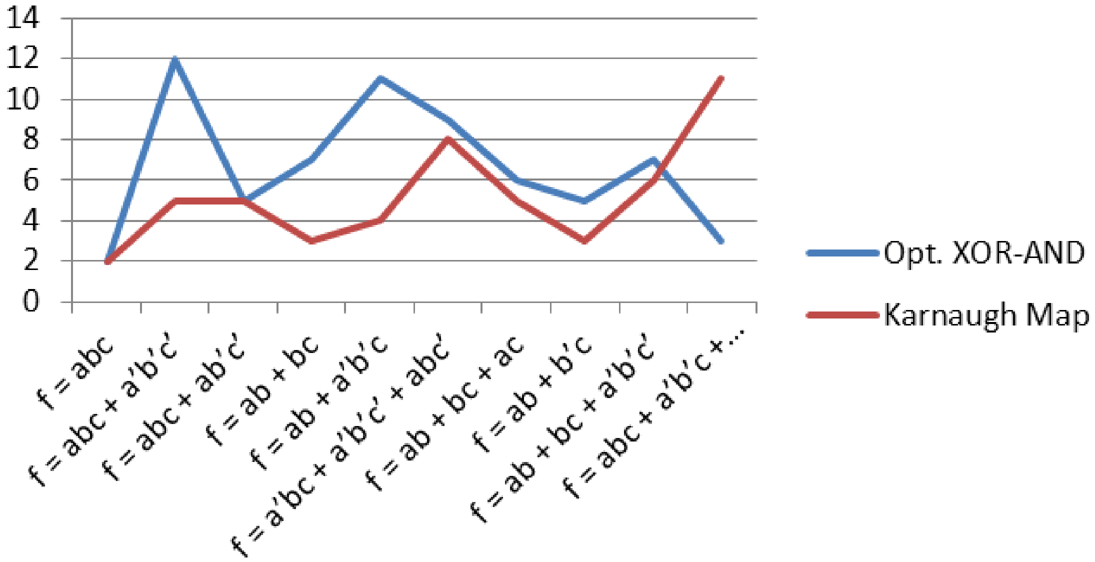


Figure 5.1. Comparative analysis – graphical representation

CHAPTER 6

TCG (Two's Complement) Gate

Ayan Chaudhuri, Mahamuda Sultana, Diganta Sengupta, Chitrita Chaudhuri and Atal Chaudhuri, "A Reversible Approach to Two's Complement Addition using a Novel Reversible TCG Gate and its 4 Dot 2 Electron QCA Architecture", Microsystem Technologies, Springer, (25), pp 1965-1975, 2018. (SCI Indexed – Impact Factor 1.581).

6 TWO'S COMPLEMENT GATE

A lot of reversible four input four output gates have been proposed in the reversible domain. The gates address some of the Boolean functions and also realize some of the fundamental Boolean gates. But thorough study of literature exhibits that a reversible gate capable of generating the 2's complement of a given four bit number has not yet been proposed. This chapter presents a 2's complement gate which produces a four bot 2's complement of a four bit number given as the input. It has been observed that each of the outputs map exactly to one and only one input thereby rendering the proposed gate as a reversible gate. The existing reversible gates which have been proposed to address specific Boolean functions are RI [**25**], R2 [**25**], TSG [**26**], HNG [**27**], SCG [**28**], RPS [**29**], HNFG [**30**] [**31**], MKG [**30**], IG [**32**], FAG [**33**], ALG [**34**], MTSG [**35**], DFG [**33**], DKG [**36**], MRG [**36**] and BVMF [**41**] gates. Proposals have also been found in terms of arithmetic sub-modules in [**71**] [**15**] [**10**] [**72**] [**12**] [**73**] with sequential circuits being proposed in [**26**] [**74**] [**75**] [**76**]. In terms of sequential logic, this chapter also presents the RS Flip Flop realization using the 2' Complement Gate (TCG). Therefore it can be assumed that sequential elements can be designed using the TCG gate.

6.1 BLOCK DIAGRAM AND TRUTH TABLE OF TCG GATE

The block diagram and the truth table is provided in Figure 6.1 and Table 6.1 respectively.

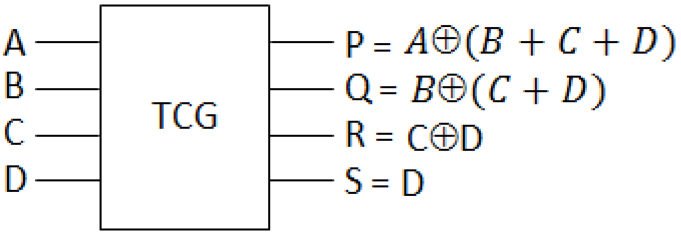
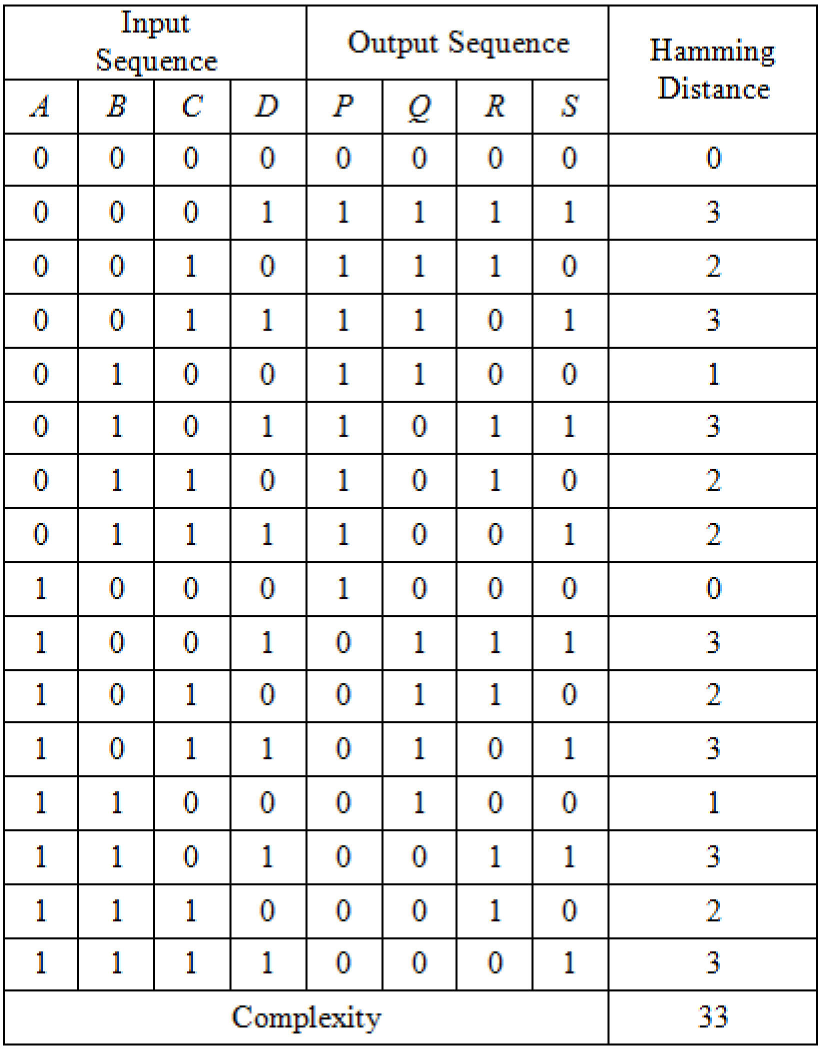


Figure 6.1. Block diagram of TCG gate

TABLE 6.1. TRUTH TABLE FOR TCG GATE

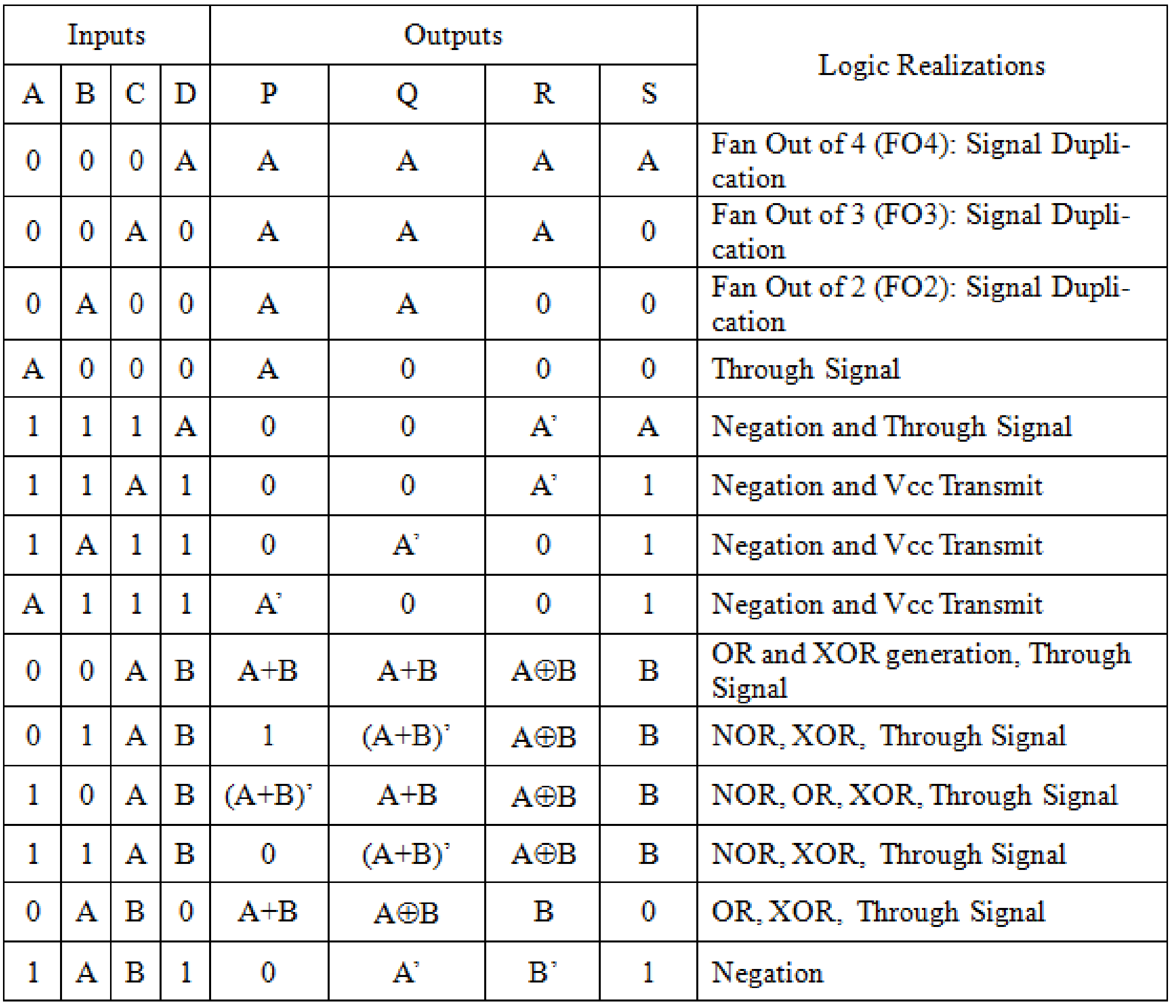


Input Sequence				Output Sequence				Hamming Distance
A	*B*	*C*	*D*	*P*	*Q*	*R*	*S*	
0	0	0	0	0	0	0	0	0
0	0	0	1	1	1	1	1	3
0	0	1	0	1	1	1	0	2
0	0	1	1	1	1	0	1	3
0	1	0	0	1	1	0	0	1
0	1	0	1	1	0	1	1	3
0	1	1	0	1	0	1	0	2
0	1	1	1	1	0	0	1	2
1	0	0	0	1	0	0	0	0
1	0	0	1	0	1	1	1	3
1	0	1	0	0	1	1	0	2
1	0	1	1	0	1	0	1	3
1	1	0	0	0	1	0	0	1
1	1	0	1	0	0	1	1	3
1	1	1	0	0	0	1	0	2
1	1	1	1	0	0	0	1	3
Complexity								33

The complexity of the TCG gate is observed to be 33 from Table 6.1. As in the case for other reversible gates, the TCG gate can also be used to realize some of the basic Boolean specifications as reflected from Table 6.2.

The concept of reversibility mandates that the fan-out from the output should be restricted to one. Using TCG Gate, and as can be seen from Table 6.2, with proper inputs, a single input can be replicated into all the four outputs. Hence this can be said to be generation of fan-out of four. It comes useful in cases where a single input requires duplication. Also from Table 6.2 it can be observed that the basic operations like NOT, OR, NOR, and XOR function can be realized by the TCG gate with specific input patterns. Since, NOR happens to be the universal Boolean gate, hence it can be claimed that the TCG gate can be cascaded to design any Boolean function. Also the XOR realization can aid the TCG gate in cryptographic applications.

TABLE 6.2. TCG GATE – BOOLEAN FUNCTION REALIZATIONS



Inputs				Outputs				Logic Realizations
A	B	C	D	P	Q	R	S	
0	0	0	A	A	A	A	A	Fan Out of 4 (FO4): Signal Duplication
0	0	A	0	A	A	A	0	Fan Out of 3 (FO3): Signal Duplication
0	A	0	0	A	A	0	0	Fan Out of 2 (FO2): Signal Duplication
A	0	0	0	A	0	0	0	Through Signal
1	1	1	A	0	0	A'	A	Negation and Through Signal
1	1	A	1	0	0	A'	1	Negation and Vcc Transmit
1	A	1	1	0	A'	0	1	Negation and Vcc Transmit
A	1	1	1	A'	0	0	1	Negation and Vcc Transmit
0	0	A	B	A+B	A+B	A⊕B	B	OR and XOR generation, Through Signal
0	1	A	B	1	(A+B)'	A⊕B	B	NOR, XOR, Through Signal
1	0	A	B	(A+B)'	A+B	A⊕B	B	NOR, OR, XOR, Through Signal
1	1	A	B	0	(A+B)'	A⊕B	B	NOR, XOR, Through Signal
0	A	B	0	A+B	A⊕B	B	0	OR, XOR, Through Signal
1	A	B	1	0	A'	B'	1	Negation

6.2 TOFFOLI GATE DESIGN FOR TCG GATE

Since the TCG gate is reversible in nature, hence it can be mapped using the fundamental reversible gate, i.e. Toffoli gate. Design using the Toffoli gate is presented in Figure 6.2. The design has been done using the Unidirectional Algorithm proposed in [**77**] [**78**].

The Toffoli Gate design for the TCG gate has been further optimized using a post – synthesis algorithm using negative control lines [**79**] [**80**] [**81**]. Figure 6.3 presents the optimized representation of TCG gate using negative control lines.

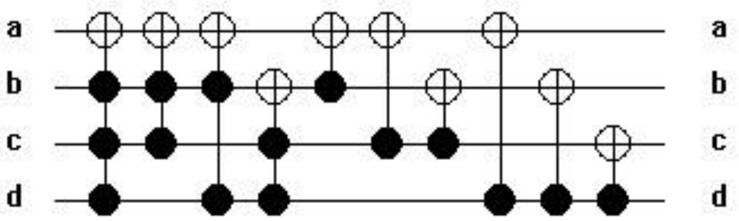


Figure 6.2. TCG gate - Toffoli representation

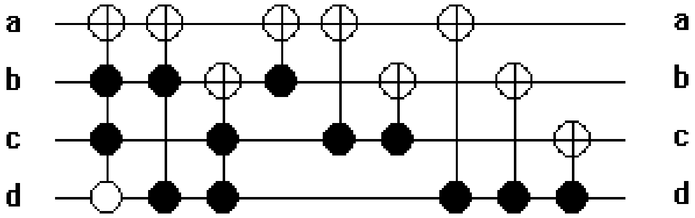


Figure 6.3. TCG gate - Optimized design using negative control lines

Figure 6.4 presents the decomposed design of the reversible Toffoli Netlist (cascade of Toffoli gates) into $Controlled - V$ and $Controlled - V^{+}$gates.

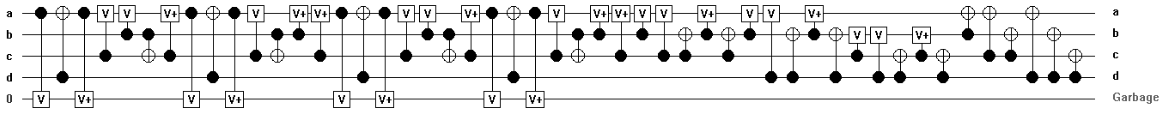


Figure 6.4. Decomposed design of Toffoli implementation presented in Figure 6.2

6.3 QCA DESIGN OF TCG GATE

The TCG gate has been implemented using QCA cells and is shown in Figure 6.5. Table 6.3 presents the majority voter count and the number of inverters required to design the TCG gate sung QCA.

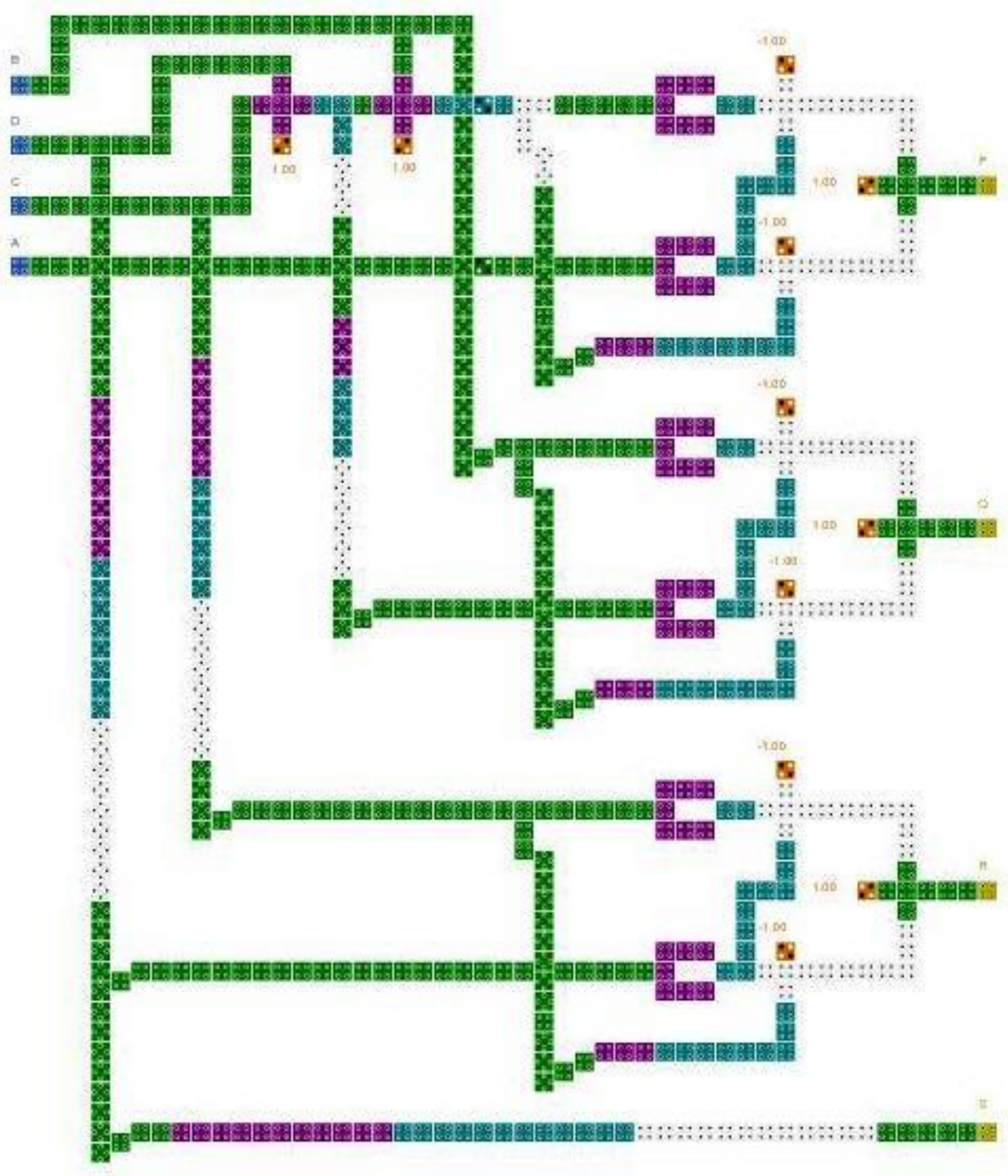
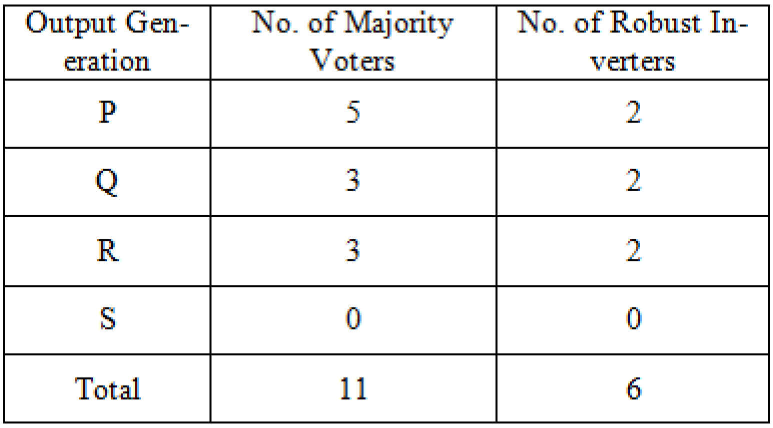


Figure 6.5. QCA design for TCG Gate

TABLE 6.3. MAJORITY VOTER COUNT AND INVERTER COUNT



Output Generation	No. of Majority Voters	No. of Robust Inverters
P	5	2
Q	3	2
R	3	2
S	0	0
Total	11	6

6.4 TWO'S COMPLEMENT ADDITION

As the name suggests, the 2's complement gate executes binary subtraction using 2's complement addition. For this purpose, another reversible gate known as the SCG gate has been used. The details for the SCG gate can be obtained from Chapter 2. The schematic for the 2's complement addition is presented in Figure 6.7. Figure 6.6 presents the input combination for which the SCG gate works as a full adder.

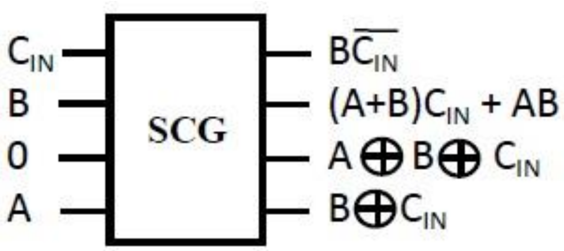


Figure 6.6. Full Adder realization using SCG gate

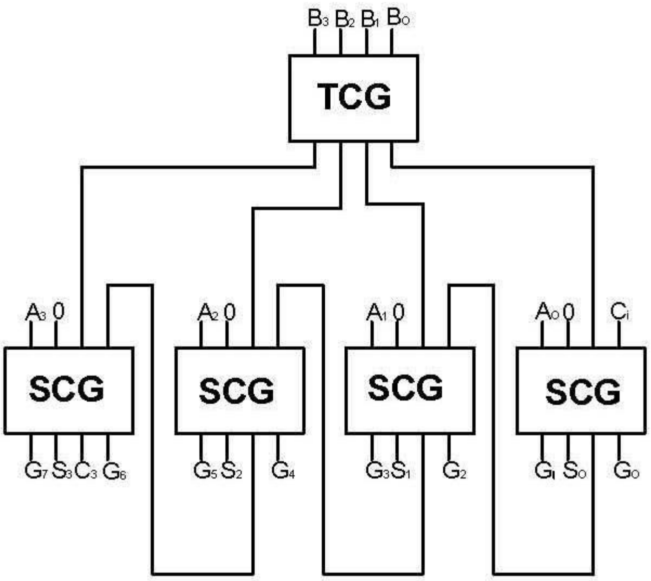


Figure 6.7. 2’s complement addition using TCG gate

6.5 RS FLIP FLOP USING TCG GATE

As discussed earlier, the TCG gate can also be used to implement the RS Flip Flop. The complete schematic for reversible RS Flip Flop is presented in Figure 6.8.

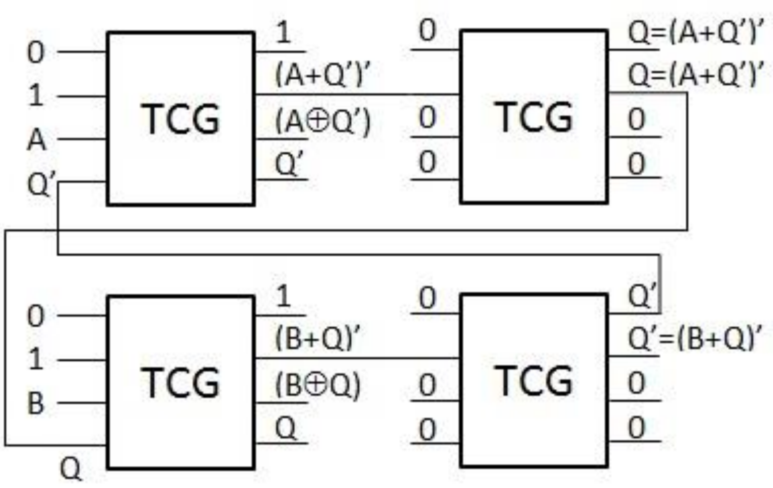


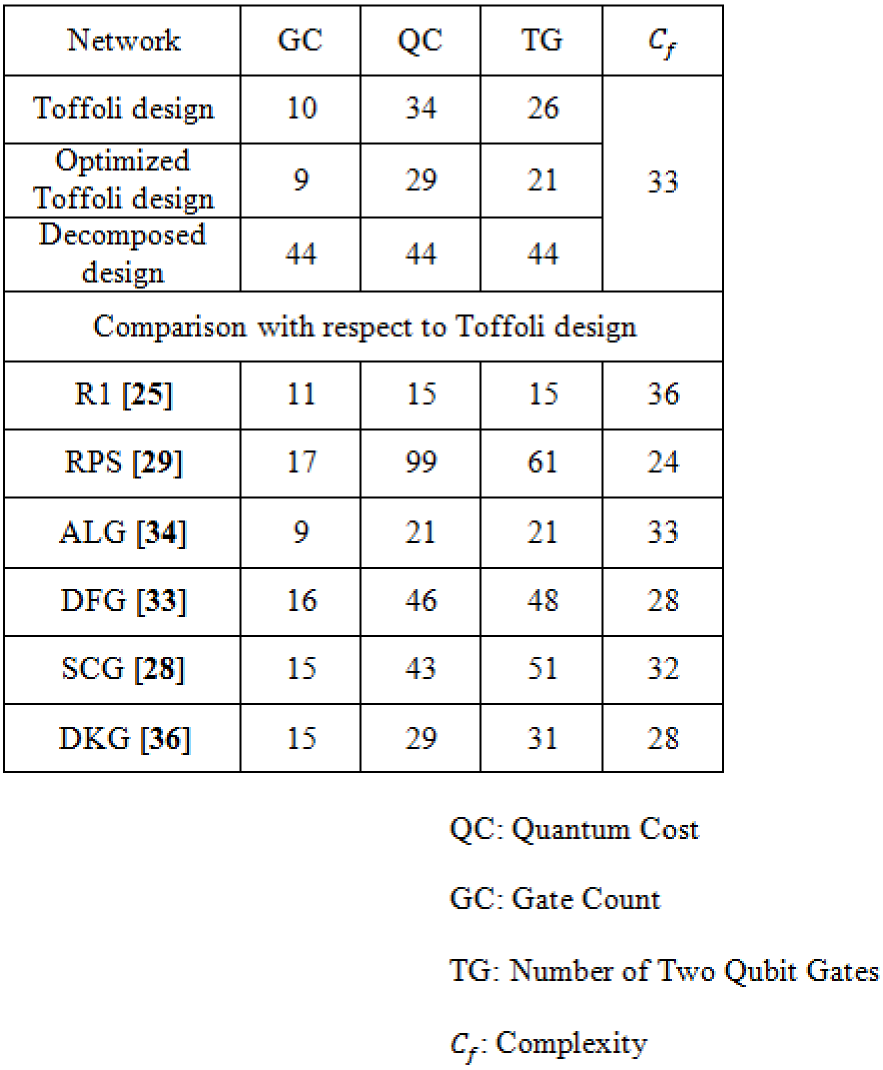
Figure 6.8. TCG gate - Reversible RS Flip Flop

The traditional design for an RS Flip Flop requires an array of NOR gates followed by output branching. The array of NOR gates are provided by the first vertical column in Figure 6.8. The second column generates a copy signal. Since, RS Flip Flop can be designed using TCG gate, hence, the TCG gate can be used to design the D Flip Flop as well as the T Flip Flop.

6.6 COMPARATIVE ANALYSIS

The comparative analysis has been done in terms of the number of Toffoli gate count, the Quantum cost, the 2-qubit gate count and the complexity. Table 6.4 presents the comparative analysis.

TABLE 6.4. COMPARATIVE ANALYSIS



Network	GC	QC	TG	C_f
Toffoli design	10	34	26	33
Optimized Toffoli design	9	29	21	
Decomposed design	44	44	44	
Comparison with respect to Toffoli design				
R1 [**25**]	11	15	15	36
RPS [**29**]	17	99	61	24
ALG [**34**]	9	21	21	33
DFG [**33**]	16	46	48	28
SCG [**28**]	15	43	51	32
DKG [**36**]	15	29	31	28

QC: Quantum Cost

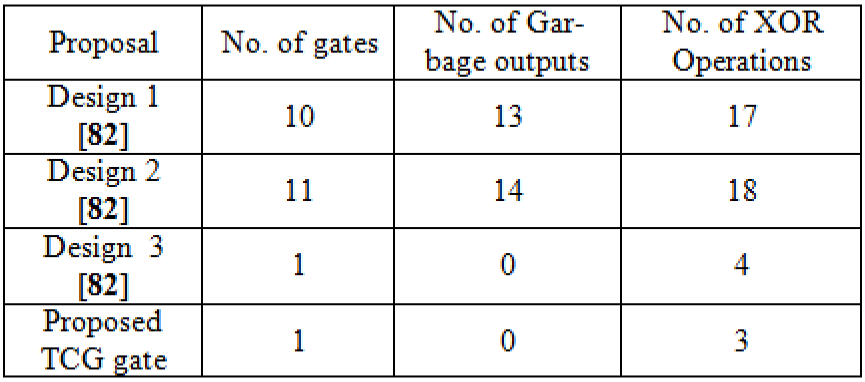
GC: Gate Count

TG: Number of Two Qubit Gates

C_f: Complexity

Table 6.4 presents the comparison with respect to another 2's complement gate designed using SSMT gate [**82**] in literature.

TABLE 6.5. PEER COMPARISON



Proposal	No. of gates	No. of Garbage outputs	No. of XOR Operations
Design 1 **[82]**	10	13	17
Design 2 **[82]**	11	14	18
Design 3 **[82]**	1	0	4
Proposed TCG gate	1	0	3

CHAPTER 7

Conclusion

7 CONCLUSION

This thesis addresses the issues regarding heat dissipation through digital circuits and faster clocking. Novel reversible circuits using QCA cells have been designed. QCA supports reversibility partially. Hence QCA can be assumed to be an inherent part of Quantum Computing. Information propagation through QCA cells occur by change in orientation in the electron positions in a cell rather than by physical travel of electrons as in classical electronics. Also the working frequency of QCA cell is in the order of THz rather than the GHz speed limit in CMOS. The first chapter details the basics of QCA cells.

7.1 CHAPTER 2 SUMMARY

Chapter 2 provides the QCA cell implementations for the four input four output reversible gates in literature. The comparisons have been done using the basic quantum metrics, area coverage of QCA designs, number of majority voters used, number of inverters used. The data in Chapter 2 can form as a reference material for future designs of reversible gates.

7.2 CHAPTER 3 SUMMARY

This chapter proposes a taxonomy for QCA research. The taxonomy has been designed using Author Keywords. The distance between the Author Keywords have been used to generate the similarity between two author keywords, later termed as taxonomy nodes. The taxonomy clusters references according to a certain node so that a researcher looking for any specific area gets hold of all the relevant references within a short time span rather than executing the time consuming enormous literature study. The taxonomy also shows that the domain of QCA holds high interest among researchers and has observed steady growth in the last five years.

7.3 CHAPTER 4 SUMMARY

Chapter 4 provides the QCA design for a Hamming Code generator and Checker circuit. QCA implementation of a XOR gate in literature has been used to design the Hamming circuits. It has been shown that the Hamming Code converter exhibits a latency of 1.25 for the Hamming Code Generator and 1.5 for the Hamming Code Checker.

7.4 CHAPTER 5 SUMMARY

Chapter 5 discusses an algorithm in literature which convert a Boolean function into XOR-AND equivalent. This chapter proposes the optimization of the XOR-AND function that has been generated. The optimization process is a library based approach. The chapter also provides a comparison of two different techniques based on XOR-AND, and K-Map techniques for QCA cell reduction.

7.5 CHAPTER 6 SUMMARY

This chapter proposes a reversible four variable gate (TCG Gate) which is capable of generating the 2’s complement of a four bit input number at the output. The gate is reversible in nature; hence each output is mapped exactly to one of the inputs. The proposed gate can also be useful in sequential realizations of Boolean functions using QCA cells because it has been shown that the proposed gate can be used to efficiently design an RS Flip Flop. Hence, since RS Flip Flop can be attained , it can be concluded that the proposed gate can find huge potential in sequential circuits.

7.6 FUTURE EXTENSION PROSPECTS

The taxonomy has been generated using the Cosine Similarity function. More taxonomy generation algorithms exist which work on Cosine Similarity function. These taxonomies can be generated and comparative analysis can be done. The XOR-AND extraction could be further extended to explore correlations between them. The 2’s complement gate can be further used to design more arithmetic structures. A lot of cryptographic algorithms exist which make use of XOR gate. Having designed the Hamming Code Generator/Checker using QCA, further exploration in cryptographic algorithms can be done in future.

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] Thomas Toffoli, "Reversible Computing," MIT Lab for Computer Science, Tech Memo MIT/LCS/TM-151, 1980.

[2] Craig S Lent, P Douglas Tougaw, and Wolfgang Porod, "Quantum Cellular Automata: The Physics of Computing with Arrays of Quantum Dot Molecules," in *Workshop on Physics and Computation, 1994. PhysComp '94*, 1994, pp. 5-13.

[3] C S Lent and P D Tougaw, "A device architecture for computing with quantum dots," *Proc. IEEE*, vol. 85, pp. 541-557, Apr 1997.

[4] C H Bennett, "Logical Reversibility of Computation," *IBM Journal of Research and Development*, vol. 17, no. 6, pp. 525 - 532, Nov 1973.

[5] Rolf Landauer, "Irreversibility and Heat Generation in the Computing Process," *IBM Journal of Research and Development*, vol. 5, no. 3, pp. 183-191, July 1961.

[6] G L Snider et al., "Experimental demonstration of quantum-dot cellular automata," *Semiconductor Science and Technology*, vol. 13, no. 1998, pp. A131-A134, 1998.

[7] F Ahmad, G M Bhat, and P Z Ahmad, "Novel Adder Circuits based on Quantum-Dot CEllular Automata (QCA)," *Circuits and Systems*, vol. 5, 142-152 2014.

[8] T J Dysart, "Modeling of Electrostatic QCA Wires," *IEEE Transactiona on Nanotechnology*, vol. 12, no. 4, pp. 553-560, July 2013.

[9] K Walus, T J Dysart, and G A Jullien, "QCADesigner: a rapid design and Simulation tool for quantum-dot cellular automata," *IEEE Transactions on Nanotechnology*, vol. 3, no. 1, pp. 26-31, March 2004.

[10] Heumpil Cho and Jr., Earl E Swartzlander, "Adder Designs and Analyses for Quantum-Dot Cellular Automata," *IEEE Transactions on Nanotechnology*, vol. 6, no. 3, pp. 374-383, May 2007.

[11] H Cho and E E Swartzlander, "Adder and Multiplier Design in Quantum-Dot Cellular Automata," *IEEE Transaction on Computers*, vol. 58, no. 6, pp. 721-727, June 2009.

[12] Stefania Perri, Pasquale Corsonello, and Giuseppe Cocorullo, "Area-Delay Efficient Binary Adders in QCA," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 22, no. 5, pp. 1174-1179, May 2014.

[13] Dariush Abedi, Ghassem Jaberipur, and Milad Sangsefidi, "Coplanar Full Adder in Quantum-Dot Cellular Automata via Clock-Zone-Based Crossover," *IEEE Transactions on Nanotechnology*, vol. 14, no. 3, pp. 497-504, May 2015.

[14] G Cocorullo, P Corsonello, F Frustaci, and S Perri, "Design of Efficient BCD Adders in Quantum Dot Cellular Automata," *IEEE Transactions on Circuits and Systems II: Express Briefs*, In Press.

[15] Vikramkumar Pudi and K Sridharan, "New Decomposition Theorems on Majority Logic for Low-Delay Adder Designs in Quantum Dot Cellular Automata," *IEEE Transactions on Circuits and Systems-II: Express Briefs*, vol. 59, no. 10, pp. 678-382, October 2012.

[16] Peng Wang, Mohammed Y Niamat, Srinivasa R Vemuru, Mansoor Alam, and Taylor Killian, "Synthesis of Majority/Minority Logic Networks," *IEEE Transactions on Nanotechnology*, vol. 14, no. 3, pp. 473-483, May 2015.

[17] V V Shende, S S Bullock, and I L Markov, "Synthesis of Quantum-Logic Circuits," *IEEE Transactions on Computer Aided Design of INtegrated Circuits and Systems*, vol. 25, no. 6, pp. 1000-1010, 2006.

[18] Kun Kong, Yun Shang, and Ruqian Lu, "An Optimized Majority Logic Synthesis Methodology for Quantum-Dot Cellular Automata," *IEEE Transactions on Nanotechnology*, vol. 9, no. 2, pp. 170-183, March 2010.

[19] Rumi Zhang, Konrad Walus, Wei Wang, and Graham A Jullien, "A Method of Majority Logic Reduction for Quantum Cellular Automata," *IEEE Transactions on Nanotechnology*, vol. 3, no. 4, pp. 443-450, December 2004.

[20] G Causapruno, M Vacca, and M Graziano, "Interleaving in Systolic-Arrays: A Throughput Breakthrough," *IEEE Transactions on Computers*, vol. 64, no. 7, pp. 1940-1953, July 2015.

[21] L Fortuna, M L Rosa, D Nicolosi, and D Porto, "Nanoscale System Dynamical Behaviors: From Quantum-Dot-Based Cell to 1-D Arrays," *IEEE Transactions on Very Large Scale INtegration Systems (VLSI) Systems*, vol. 12, no. 11, pp. 1167-1173, NOvember 2004.

[22] R K Kummamuru et al., "Operation of a Quantum-Dot Cellular Automata (QCA) Shift Register and Analysis of Errors," *IEEE Transactions on Electron Devices*, vol. 50, no. 9, pp. 1906-1913, September 2003.

[23] Baris Taskin and Bo Hong, "Improving Line-Based QCA Memory Cell Design Through Dual Phase Clocking," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 16, no. 12, pp. 1648-1656, December 2008.

[24] M Crocker, X S Hu, M Niemier, M Yan, and G Bernstein, "PLAs in Quantum-Dot Cellular Automata," *IEEE Transactions on Nanotechnology*, vol. 7, no. 3, pp. 376-386, May 2008.

[25] Dilip P. Vasudevan, Parag K. Lala, Jia Di, and J. Patrick Parkerson, "Reversible-Logic Design With Online Testability," *IEEE Transactions On Instrumentation And Measurement*, vol. 55, no. 2, pp. 406 - 414,

April 2006.

[26] H Thapliyal and M B Srinivas, "Novel Reversible `TSG' Gate and Its Application for Designing Components of Primitive Reversible/Quantum ALU," in *Fifth International Conference on Information, Communications and Signal Processing, 2005* , 2005.

[27] Goutam Kumar Maity and Santi P Maity, "Implementation of HNG using MZI," in *Third International Conference on Computing Communication & Networking Technologies (ICCCNT), 2012* , 2012, pp. 1-6.

[28] Diganta Sengupta, Mahamuda Sultana, and Atal Chaudhuri, "Realization of a Novel Reversible SCG Gate and its Application for Designing Parallel Adder/Subtractor and Match Logic," *International Journal of Computer Applications*, vol. 31, no. 9, pp. 30-35, October 2011.

[29] Rekha K James, K Poulose Jacob, and Sreela Sasi, "Design of compact reversible decimal adder using RPS gates," in *World Congress on Information and Communication Technologies (WICT), 2012*, 2012, pp. 344 - 349.

[30] Majid Haghparast and Keivan Navi, "A Novel Reversible Full Adder Circuit for Nanotechnology Based Systems," *Journal of Applied Sciences*, vol. 7, no. 24, pp. 3995-4000, 2007.

[31] Majid Haghparast and Keivan Navi, "A Novel Reversible BCD Adder For Nanotechnology Based Systems," *American Journal of Applied Sciences*, vol. 5, no. 3, pp. 282-288, 2008.

[32] Md. Saiful Islam, Muhammad Mahbubur Rahman, and Zerina Begum, "Fault tolerant reversible logic synthesis: Carry look-ahead and carry-skip adders," in *International Conference on Advances in Computational Tools for Engineering Applications, 2009. ACTEA '09.*, 2009, pp. 396-401.

[33] S B Rashmi, T G Umarani, and H K Shreedhar, "Optimized Reversible Montgomery Multiplier ," *International Journal of Computer Science and Information Technologies*, vol. 2, no. 2, pp. 701-706, 2011.

[34] M Arun and S Saravanan, "Reversible Arithmetic Logic Gate (ALG) for Quantum Computation," *International Journal of Intelligent Engineering and Systems*, vol. 6, no. 3, pp. 1-9, 2013.

[35] Ashis Kumer Biswas, Md. Mahmudul Hasan, Ahsan Raja Chowdhury, and Hafiz Md. Hasan Babu, "Efficient approaches for designing reversible Binary Coded Decimal adders," *Microelectronics Journal, Elsevier*, vol. 39, no. 12, pp. 1693-1703, December 2008.

[36] Papiya Biswas, Namit Gupta, and Nilesh Patidar, "Basic Reversible Logic Gates and It’s QCA Implementation," *Int. Journal of Engineering Research and Applications*, vol. 4, no. 6, pp. 12-16, June 2014.

[37] S Banerjee, A K Pal, M Sultana, D Sengupta, and A Das, "Reversible Code Converters based on Application Specific Four Variable Reversible Gates," in *International Conference on Emerging Technologies in Data Mining and Information Security (IEMIS 2018), Springer*, Kolkata, 2018, p. In Press.

[38] M Sultana, A Chaudhuri, D Sengupta, and A Chaudhuri, "Logic Design and Quantum Mapping of a Novel Four Variable Reversible s2c2 Gate," in *Annual Convention of Computer Society of India (CSI 2017), Springer*, Kolkata, 2018, pp. 416-427.

[39] A Chaudhuri, M Sultana, D Sengupta, and A Chaudhuri, "A novel reversible two's complement gate (TCG) and its quantum mapping," in *Devices for Integrated Circuit (DevIC), IEEE*, Kalyani, 2017, pp. 252-256.

[40] A Chaudhuri, M Sultana, D Sengupta, C Chaudhuri, and A Chaudhuri, "A Reversible Approach to Two's Complement Addition using a Novel Reversible TCG Gate and its 4 Dot 2 Electron QCA Architecture," *Microsystem Technologies, Springer*, vol. 25, pp. 1965-1975, July 2018.

[41] H R Bhagyalakshmi and M K Venkatesha, "Design of a Multifunction BVMF Reversible Logic Gate and its Applications," *International Journal of Computer Applications*, vol. 32, no. 3, pp. 0975 – 8887, October 2011.

[42] A Chaudhuri, M Sultana, D Sengupta, D De, and C Chaudhuri, "An empirical study on published literature in QCA – A Taxonomy Proposal," *IET Computers & Digital Techniques*, Communicated.

[43] A Chaudhuri, D Sengupta, and C Chaudhuri, "Adiabatic Hamming Code Generator-Checker using 4-Dot-2-Electron Quantum Dot Cellular Automata," *International Journal of Engineering and Advanced Technology*, vol. 8, no. 6, pp. 4435-4439, August 2019.

[44] A Chaudhuri, M Sultana, D Sengupta, C Chauhduri, and A Chaudhuri, "A Comparative Analysis of CNF and XOR-AND Representations for QCA Majority Gate Estimation," in *Annual Convention of Computer Society of India (CSI 2017), Springer*, Kolkata, 2018, pp. 402-415.

[45] M Sultana, A Chaudhuri, D Sengupta, and A Chaudhuri, "Toffoli Netlist and QCA Implementations for Existing Four Variable Reversible Gates – A Comparative Analysis," *Microsystem Technologies, Springer*, vol. 25, pp. 1987-2009, July 2018.

[46] M J Grant, "Key words and their role in information retrieval," *Health Information and Libraries Journal*, vol. 27, no. 3, pp. 173-175, September 2010.

[47] R L Cilibrasi and P M B Vitanyi, "The google similarity distance," *IEEE Transactions on Knowledge and Data Engineering*, vol. 19, no. 3, pp. 370-383, 2007.

[48] S L Camina, "A Comparison of Taxonomy Generation Techniques

Using Bibliometric," Massachusetts Institute of technology, EECS Thesis 2010.

[49] P H Raven, B Berlin, and D E Breedlove, "The Origins of Taxonomy," *Science*, vol. 174, no. 4015, pp. 1210-1213, 1971.

[50] C S Lent, P D Tougaw, W Porod, and G H Bernstein, "Quantum cellular automata," *Nanotechnology*, vol. 4, no. 1, pp. 49-57, 1993.

[51] Craig S Lent and Beth Isaksen, "Clocked Molecular Quantum-Dot Cellular Automata," *IEEE Transactions on Electron Devices*, vol. 50, no. 9, pp. 1890-1896, September 2003.

[52] R Landauer, "Irreversibility and Heat Generation in the Computing Process," *IBM Journal of Research and Development*, vol. 5, no. 3, pp. 183 - 191, July 1961.

[53] H Rashidi and A Rezai, "High-performance full adder architecture in quantum-dot cellular automata," *The Journal of Engineering* , vol. 2017, no. 7, pp. 394-402, 2017.

[54] T N Sasamal , A K Singh, and U Ghanekar , "Efficient design of coplanar ripple carry adder in QCA," *IET Circuits, Devices & Systems*, vol. 12, no. 5, pp. 594-605, September 2018.

[55] Y Zhang, G Xie, M Sun, and H Lv, "Design of normalised and simplified FAs in quantum-dot cellular automata," *The Journal of Engineering* , vol. 2017, no. 10, pp. 557-565, October 2017.

[56] T Sadhu, B Das, D De, and J C Das, "Design of binary subtractor using actin quantum cellular automata," *IET Nanobiotechnology*, vol. 12, no. 1, pp. 32-39, February 2018.

[57] B Debnath, J C Das, and D De, "Reversible logic-based image steganography using quantum dot cellular automata for secure nanocommunication," *IET Circuits, Devices & Systems*, vol. 11, no. 1, pp. 58-67, January 2017.

[58] D Sengupta and M Sultana, "Taxonomy of Decimal Multiplier Research," in *Conference on Algorithms and Applications (ALAP 2018), Springer*, Kolkata, 2018, pp. 3-21.

[59] M Sultana, D Sengupta, and A Chaudhuri, "Paper Title: Taxonomy Proposal for Research on Reversible Logic," *International Journal of Engineering and Advanced Technology*, vol. 8, no. 6, August 2019.

[60] A K Pal, S Banerjee, N Dey, and D Sengupta, "IoT Based Home Automation - A Proposal for Taxonomy," in *IEEE 3rd International Conference for Convergence in Technology (I2CT)*, Pune, 2018, pp. 1-6.

[61] D Sengupta, "Taxonomy on Ambient Computing – A Research Methodology Perspective," *International Journal of Ambient Computing and Intelligence*, vol. 11, no. 1, In Press 2019.

[62] C S Lent, P D Tougaw, and W Porod, "Bistable saturation in coupled quantum dots for quantum cellular automata," *Appl. Phys. Lett.*, vol. 62, p. 714, 1993.

[63] G L Snider et al., "Quantum-dot cellular automata: Review and recent experiments," *Quantum-dot cellular automata: Review and recent experiments*, vol. 85, no. 8, pp. 4283-4285, April 1999.

[64] S Angizi, E Alkaldy, N Bagherzadeh, and K Navi, "Novel Robust Single Layer Wire Crossing Approach for Exclusive OR Sum of Products Logic Design with Quantum-Dot Cellular Automata," *Journal of Low Power Electronics*, vol. 10, pp. 256–271, 2014.

[65] M Mohammadi, S Gorgin, and M Mohammadi, "Design of non-restoring divider in quantum-dot cellular automata technology," *IET Circuits, Devices & Systems*, vol. 11, no. 2, pp. 135–141, 2017.

[66] NURIDDINSAFOEV, J S LEE, and J C JEON, "QCA XOR GATE FOR ARITHMETIC AND LOGIC CIRCUIT DESIGN," *International Journal of Advanced Computational Engineering and Networking*, vol. 5, no. 7, pp. 19-22, July 2017.

[67] M Ziabari, A M Kassai, A Ziabari , and S E Maklavani, "Designing a Hamming Coder/Decoder Using QCAs," *Journal of Applied Sciences*, vol. 8, pp. 2569-2576, 2008.

[68] G Xie, Y Xiang, Y Zhang, S Liu, and H LV, "Design and Implementation of Encoding and Check Code Circuit with Hamming Code on QCA," *International Journal of Unconventional Computing*, vol. 10, no. 5/6, pp. 391-404, 2014.

[69] A Ahmadpour, A A Sha, and M Ziabari, "A Novel formulation of Hamming Code," in *IEEE 6th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology*, Pattaya, Thailand, 2009.

[70] O Dajani, "Emerging Design Methodology And Its Implementation Through RNS And QCA," Wayne State University, Dissertations 646, 2013.

[71] Vikramkumar Pudi and K Sridharan, "Efficient Design of a Hybrid Adder in Quantum-Dot Cellular Automata," *IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS*, vol. 19, no. 9, pp. 1535-1548, September 2011.

[72] H Cho and Earl E Swartzlander, "Adder and Multiplier Design in Quantum-Dot Cellular Automata," *IEEE Transaction on Computers*, vol. 58, no. 6, pp. 721-727, June 2009.

[73] Liang Lu, Weiqiang Liu, Maire O'Neill, and Earl E Swartzlander Jr., "QCA Systolic Array Design," *IEEE Transactions on Computers*, vol. 62, no. 3, pp. 548-560, March 2013.

[74] Himanshu Thapliyal, Nagarajan Ranganathan, and Saurabh Kotiyal,

"Design of Testable Reversible Sequential Circuits," *IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS*, vol. 21, no. 7, pp. 1201-1209, July 2013.

[75] MIN-LUN CHUANG and CHUN-YAO WANG, "Synthesis of Reversible Sequential Elements," *ACM Journal On Emerging Technologies In Computing Systems*, 2008.

[76] HIMANSHU THAPLIYAL and NAGARAJAN RANGANATHAN, "Design of Reversible Sequential Circuits Optimizing Quantum Cost, Delay, and Garbage Outputs," *ACM Journal on Emerging Technologies in Computer Systems*, vol. 6, no. 4, Dec 2010.

[77] D M Miller, D Maslov, and G W Dueck, "A transformation based algorithm for reversible logic synthesis," in *Design Automation Conference*, 2003, pp. 318 - 323.

[78] Dmitri Maslov, Gerhard W Dueck, and D Michael Miller, "Synthesis of Fredkin–Toffoli Reversible Networks," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 13, no. 6, pp. 765-769, June 2005.

[79] D Maslov and G W Dueck, "Level Compaction iin Quantum Circuits," in *IEEE Congress on Evolutionary Computing*, 2006, pp. 2405-2409.

[80] K Datta et al., "Exploiting Negative Control Lines in the Optimization of Reversible Circuits," in *International Conference on Reversible Computing*, 2013, pp. 209-220.

[81] Kamalika Datta, Indranil Sengupta, and Hafizur Rahaman, "A Post-Synthesis Optimization Technique for Reversible Circuits Exploiting Negative Control Lines," *IEEE Transactions on Computers*, vol. 64, no. 4, pp. 1208 - 1214, April 2015.

[82] Vandana Shukla, O P Singh, G R Mishra, and R K Tiwari, "Design of a 4-bit 2’s Complement Reversible Circuit for Arithmetic Logic Unit Applications," in *The International Conference on Communication, Computing and Information Technology (ICCCMIT) 2012*, 2012, pp. 1-5.