IJFANS INTERNATIONAL JOURNAL OF FOOD AND NUTRITIONAL SCIENCES
ISSN PRINT 2319 1775 Online 2320 7876
 Journal Volume 10, Iss 11, 2021

IMPLEMENTATION OF MULTI-BIT MULTIPLEXER USING MAJORITY LOGIC BASED QCA

[1]Aade Kailas Ukala,[2]A.Vinitha,[3]S.Ravinder,[4]E.Ruchitha
[1234]Assistant Professor Department Of ECE
Kshatriya College of Engineering

Abstract:

Boolean logic functions is possible to realize using collisions in bio-molecular quantum automata systems. Actin is a thin muscle filament consisting of two protein chains that mainly contribute to animal nervous system and eukaryotic cellular structure. Actin quantum cellular automata is also capable of quantum computation with help of qubits, if superposition of states is employed in the input and conducting moves using rules that are applicable to each of the actin molecules. In the present work, design of multiplexers like Multi bit multiplexer using actin filaments are proposed. For the sake of simplicity, classical bit representation is used to realize logic gates like AND gate, NOT gate and OR gate with the quantum automata. Finally these gates are utilized to design the proposed multiplexer. By comparing the efficiency of methodologies used in designing the circuits, optimized design for the proposed multiplexer is suggested.

I.INTRODUCTION

1.1 OVERVIEW

Optimizations in VLSI have been done on three factors: Area, Power and Timing (Speed).Area optimization means reducing the space of logic which occupy on the die. This is done in both front-end and back-end of design. In front-end design, proper description of simplified Boolean expression and removing unused states will lead to minimize the gate/transistor utilization. Partition, Floor planning, Placement, and routing are perform in back-end of the design which is done by CAD tool .The CAD tool have a specific algorithm for each process to produce an area efficient design similar to Power optimization. Power optimization is to reduce the power dissipation of the design which suffers by operating voltage, operating frequency, and switching activity. The first two factors are merely specified in design constraints but switching activity is a parameter which varies dynamically, based on the way which designs the logic and input vectors. Timing optimization refers to meeting the user constraints in efficient manner without any violation otherwise, improving performance of the design.

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

Quantum dots are semiconductors confined in all three dimensions of space or alternatively, it can be noted that Quantum dot is a simple charge container and it is three dimensionally confined [8]. The promising alternative of CMOS paradigm is the Quantum dot cellular Automata (QCA) which is used to represent the information in binary M’ and binary ‘O’ in terms of electronic charge configuration. In 1993, C. S. Lent et al. first introduced the theoretical Quantum dot Cellular Automata [3] and in early 1999, C. S. Lent et al. described the experimental approach to design QCA cell with GaAs [8]. The dynamic behaviour of QCA was discussed with the help ofthe hart tree approximation [4], Quantum mechanics is also involved in finding out the cell size and dot radius of a single QCA cell. Hence, QCA became research interest to establish as

strong CMOS alternative. During last decades, in nanotechnology era, an exhaustive research has been carried out in this domain. QCA is still in infancy stage, needs lots ofstudy for QCA logic circuit design. The low power reversible logic circuit design, tile based logic circuit design as well as its defect analysis are prime problem domain. The ternary computing with QCA is most challenging task in this domain since no such improvement is noticed. The multivalued computing, specifically ternary computing is an emerging domain of research due the potential advantages like greater data storage capability, faster arithmetic operations, better support for numerical analysis, application ofnon-deterministic and heuristic procedures, communication protocol and effective solution for non-binary problems. Nano-scale logic circuit fabrication is suffering from defects that may occur during fabrication. It is also noticed that QCA fabrication is suffering from high probability of defect. It was reported in several proposals [59-60, 63-70] that defects are considered mainly on deposition phase. It is assumed that successful chemically synthesized QCA cells are deposited on the substrate. In this phase three common defects that had been analyzed are (a) extra cell deposition i.e. extra QCA cell/s is/are deposited than the original requirement of cell arrangement, (b) missing cell deposition/ un-deposited cell deposition, i.e. the QCA cell/s is/are not deposited as required in original design, (c) displaced /misplaced cell deposition, i.e. QCA cell/s is/are misplaced from the exact position of deposition. These three types of defects may cause major 2 fatal errors in QCA manufacturing. The device or gate design using QCA required a permissible defect tolerance on the above-mentioned defects such that the device no longer loses its characteristics. Hence, defect analysis is becoming most promising problem domain in QCA

1.2 PROBLEM STATEMENT

It is well known that the Complementary Metal Oxide Semiconductor (CMOS) technology based digital computers conceived two innovative ideas, in one idea information are represented with binary ‘O’ and binary ‘ 1 ’ and another one is that electronic charge state is used to represent the information in terms of current switch. The CMOS provides micro scale computing with high density and low power Large Scale Integrated Circuit (VLSI), However, such technology was found to have several drawbacks like high leakage of current, power dissipation in terms of heat, and limitation of speed in GHz range. Moreover, this technology has arrived at its limitation as per Moore’s Law, i.e., unit cost is shrinking as number of circuit components rises. Every eighteen months number of circuit components become double [1] as well as the industry is now facing an increasing important trends of “More-than-Moore”, reported in the Semiconductor Industries Association’s International Roadmap for Semiconductors [2]. Researchers have to find out a strong alternative of CMOS technology for VLSI design. Nanotechnology was found as a strong alternative, subject to some confusion and controversy and complicated by the fact that there are naturally occurring nano size materials and other nanosize particle, in the range from 1pm down to 10A. Nanotechnologies won its existence in development within field ofmicroelectronics. Nanomechanical computing elements are scalable in terms ofinput size and depth ofpropagation path analyzed using a bounded continuum model. Boolean logic functions ofNOT, AND, OR, and XOR are realized. Nanotechnology should not be viewed as a single technology that only affects the specific area. It compensates the limitation in many existing technology. Quantum dot Cellular Automaton (QCA) is an emerging research domain in nanotechnology [3-10].

1.3 MOTIVATION

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

is constructed from four quantum dots arranged in a square pattern. These quantum dots are sites electrons can occupy by tunneling to them.

II. ESSENTIALS OF QUANTUM-DOT CELLULAR AUTOMATA

2.1 Introduction

This chapter will provide the background material needed for a full and complete discussion of the work to be presented in this thesis. The chapter begins with a discussion of the QCA device and then extends to logical circuits that are constructed from the basic QCA device. The chapter includes the discussion on how QCA devices are clocked. Then the chapter concludes with the different models of the QCA.

2.2 QCA Device Background

QCA cells perform computation by interacting coulombically with neighboring cells to influence each other's polarization. In the following subsections we review some simple, yet essential, QCA logical devices: a majority gate, QCA wires, and more complex combinations of QCA cells.

2.2.1 The Basic QCA Device

The basic device in QCA is a QCA cell which enables both the computation and transmission of the information. A QCA cell consists of a hypothetical square space with four electronic sites and two electrons. The electronics sites, called Dots, represent the locations which the electrons can occupy. The dots are coupled through quantum mechanical tunneling barriers and electrons can tunnel through them depending on the state of the system. Exactly two mobile electrons are loaded in the cell and can move to different quantum dots in the QCA cell by means of electron tunneling. Tunneling paths are represented by the lines connecting the quantum dots in Figure 2.1. Coulombic repulsion will cause the electrons to occupy only the corners of the QCA cell resulting in two specific polarizations [122]. Electron tunneling is assumed to be completely controllable by potential barriers (that would exist underneath the cell) that can be raised and lowered between adjacent QCA cells by means of capacitive plates.

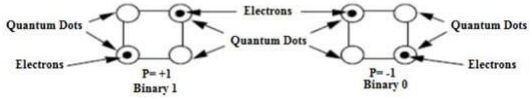


Figure 2.1. QCA cell polarizations and representations of binary 1 and binary 0.

For an isolated cell there are two energetically minimal equivalent arrangements of the two electrons in the QCA cell, denoted cell polarization P = +1 and cell polarization P = -1. Cell polarization P = +1 represents a binary 1 while cell polarization P = -1 represents a binary 0. This concept is also illustrated graphically in Figure 2.1. It is also worth noting that there is an unpolarized state as well. In an unpolarized state, inter-dot potential barriers are lowered which reduces the confinement of the electrons on the individual quantum dots. Consequently, the cells exhibit little or no polarization and the twoelectron wave functions have delocalized across the cell [133] as shown in Figure 2.2

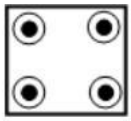


Figure 2.2. QCA unpolarized cell.

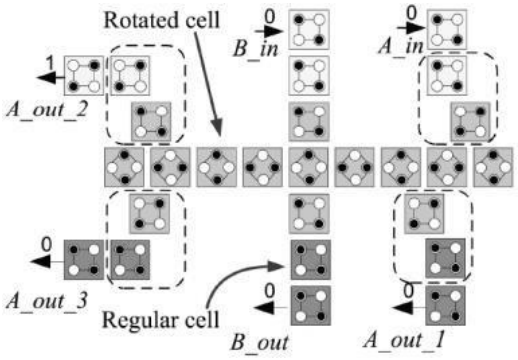
The numbering of dots denoted by i in the cell goes clockwise starting from the dot on the top right with i =1, bottom right dot i =2, bottom left dot i =3, and top left dot i=4. The polarization P in a cell is defined as Where Pi denotes the electronic charge at dot i. The polarization measures the charge configuration i.e. the extent to which the electronic charge is distributed among the four dots [3] [90].

2.2.4. 90-Degree QCA Wire

Figure 2.5 illustrates how a binary value propagates down the length of a QCA wire [129]. A 90-degree wire is a horizontal row of QCA cells. The binary signal propagates from left-to-right because of the Coulombic interactions between cells.

In Figure 2.5. Binary '0' (from polarization P = -1) will propagate down the length of the wire because of the Coulombic interaction between the cells. Initially, the electron repulsion caused by Coulombic interaction between cell 1 and cell 2 will cause cell 2 to change polarizations. Then, the electron repulsion between cell 2 and cell 3 will cause cell 3 to change polarizations. This process will continue down the length of the QCA wire.

ISSN PRINT 2319 1775 Online 2320 7876
 Journal Volume 10, Iss 11, 2021



III. LITERATURE SURVEY

The evolution of electronic information technology (IT) and communications has been mainly possible by continuous progress in silicon-based Complementary Metal Oxide Semiconductor (CMOS) technology. This continuous progress has been maintained mostly by its dimensional scaling, which results in exponential growth in both device density and performance. The reduction in costperfunction has steadily been increasing the economic productivity with every new technology. In addition to its scalability, the unique device properties such as high input resistance, self - isolation, zero static power dissipation, simple layout and process steps have made CMOS transistors as the main components of CMOS integrated circuits (ICs). However,the dimensions of CMOS transistor shrinks and approaches towards the close proximity between source and drain, which reduces the ability of the gate electrode to control the potential distribution and the flow of current in the channel region. Thus avoiding further reduction size.

Dimensional scaling of CMOS transistors is reaching their fundamental physical limits [24-25]. Therefore, research has been actively carried out to find an alternative way to continue to follow Moore's law. Among these efforts, various kinds of alternative memory and logic devices, so called "Beyond CMOS Devices," have been proposed [8]. These nano-devices take advantage of the quantum mechanical phenomena and ballistic transport characteristics under lower supply voltage and hence low power consumption. These devices are expected to be used for ultra-high density integrated electronic computers due to their extremely small size.

Nano-wire Field-Effect Transistors (NWFETs) have drawn promising attention and have been considered an alternative to continue CMOS scaling, since their nonplanar geometry provides superior electrostatic control of the channel than the conventional counter parts. The increasing attention in Nano-wire research stems from several key factors; their cost-effective "bottom–up" fabrication and high-yield reproducible electronic properties [26-28], which pave way for some fabrication challenges, higher carrier mobility, smooth surfaces and the ability to produce radial and axial Nano-wire hetero-structures [29-30], better scalability resulting from the fact that diameter of Nano-wires can be controlled down to well below 10 nm [31-32]. However, due to their smaller diameters, the inversion charge changes from surface inversion to bulk inversion due to quantum confinement. Thus, variations in Nano-wire dimensions due to fabrication imperfections can lead to perturbations in the carrier potential and scattering that degrade the charge transport characteristics. Also, variations in Nano-wire diameters may lead to a variation in FET threshold voltage. Reducing variability is therefore a key challenge in making Nano-wire FETs a viable technology. Furthermore, quantum confinement effects make modeling of Nano-wire transistors a complex problem. The physics related to the operation of Nano-wire transistors needs to be well articulated so that simple compact models, including ballistic transport and realistic sub band parameters, can be developed for circuit design using SPICE-like simulators [33].

IV. PROPOSED METHOD

4.1. Introduction

Quantum-dot cellular Automata is an emerging technology which provides the various advantages such as faster speed, smaller size, and low power consumption etc. The fundamental device is a QCA cell and can be used to design the various types of circuits (Combinational and sequential). Thus it is a fundamental block of nano-electronic circuits. This chapter extends the design of QCA based combinational circuits and provides an extensive analysis of proposed designs. This chapter discusses the various types of combinational circuits which include the Adder, Subtractor, Multiplexers, Encoders, Code

IJFANS INTERNATIONAL JOURNAL OF FOOD AND NUTRITIONAL SCIENCES
ISSN PRINT 2319 1775 Online 2320 7876
 Journal Volume 10, Iss 11, 2021

converters etc. In this work all the circuits have been designed using the majority gates.

Modular Design of Multiplexers

An electronic multiplexer allows a system to select one of the several input signals and forward it to the output. This capability makes it possible to use it as a switch. Thus the multiplexer is an extremely important part of signal control systems, because it allows the system to choose one of several inputs to be forwarded to the output. The signal selection to be forwarded to the output of the multiplexer is made by the selection lines.

2:1 Multiplexer

The characteristic equation of 2:1 multiplexer is given as: BS (4.11)+ AS =F where A and B are the multiplexer signal inputs, S corresponds to the selector input and F is the multiplexer output signal. The equation (4.11) has two product terms added together, thus we require three majority logic gates in which two performs the AND operation and the one will provide the OR operation. The schematic and majority logic representation of 2:1 multiplexer are shown in Figure. .

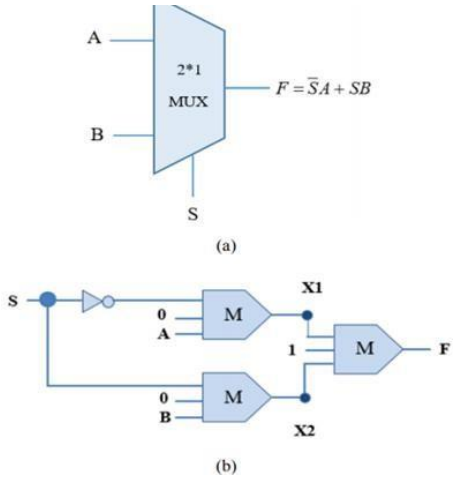
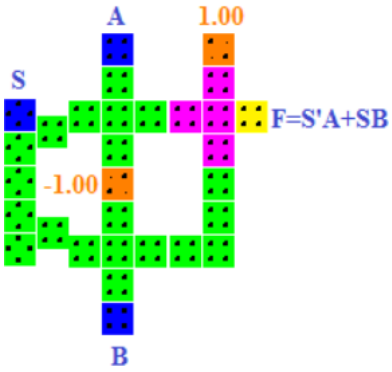


Fig:(a) Schematic of 2:1 MUX (b) MV representation of 2:1 MUX

The QCA implementation and the simulation output of the 2:1 multiplexer are shown in Figure 4.17 & Figure 4.18 respectively.



In the proposed design of 2:1 multiplexer, selection line (S) is implemented with the help of 45° rotated QCA cells. The advantage of using such line for signal propagation is that wires can be crossed on the same level without interference or crosstalk. Furthermore, because of the alternating polarization, both the signal and its complement are easily extracted. The another advantage of this design is that it requires the minimum clock zones, lesser cells and occupies less area. The comparison of the various 2: 1 multiplexer are illustrated in Table.

Structure	Parameters			
2: 1 Mux	Cell Count	Cell Area (μm^2)	Total Area (μm^2)	Latency (Clock Zones)
Mardiris et al. [107]	62	0.020	0.090	4
Mukhopadhyay et al. [165]	49	0.015	0.066	2
Askari et al. [166]	34	0.011	0.050	4
Hashemi et al.[167]	38	0.012	0.037	4
Proposed 2: 1 Mux	**30**	**0.009**	**0.04**	**2**

Fig:Comparison of various 2:1 Multiplexers

4:1 Multiplexer

A 4:1 can be realized with by the modular design of 2:1 multiplexer. The characteristic equation of 4:1 multiplexer can be realized as: D.S0.S1 (4.16)+C.S0.S1+ B.S0.S1+ A.S0.S1=F Where A, B, C, D corresponds the multiplexer inputs and S0, S1 represents the selection lines. The schematic and the majority logic representation of the 4:1 multiplexer are shown in Figure.

IJFANS INTERNATIONAL JOURNAL OF FOOD AND NUTRITIONAL SCIENCES
ISSN PRINT 2319 1775 Online 2320 7876
 Journal Volume 10, Iss 11, 2021

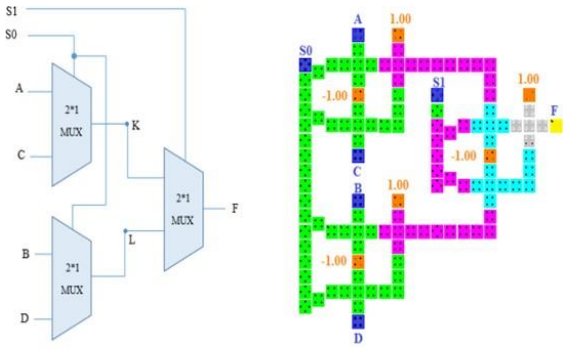


Fig: (a) Schematic of 4:1MUX (b) QCA layout of 4:1 Multiplexer.

As 4:1 multiplexer is obtained by using 2:1 multiplexer, and only thee majority gate are required for the implementation of 2:1 multiplexer. Therefore, nine majority gates are required for implementing the 4:1 multiplexer. The simulation output of the 4:1 multiplexer is given in Figure.

The comparison of various 4:1 multiplexers is given in Table 4.4

Structure	Parameters			
4: 1 Mux	**Cell Count**	**Cell Area (μm^2)**	**Total Area (μm^2)**	**Latency (Clock Phases)**
Mardiris et al. [107]	215	0.069	0.25	6
Mukhopadhyay et al. [165]	166	0.053	0.27	4
Askari et al. [166]	127	0.041	0.22	8
Hashemi et al. [168]	134	0.043	0.19	5
Proposed 4: 1 Mux	**119**	**0.038**	**0.18**	**4**

Table 4.4: Comparison of 4:1 Multiplexers

8:1 Multiplexer

The modular approach is also adopted for designing the 8:1 multiplexer, and it only requires seven 2:1 multiplexers. The Schematics of 8:1 multiplexer is shown in Figure 4.21(a). The three selection lines S0, S1 and S2 are used to select the appropriate input. Depending on the bit status of these selection lines an appropriate input is selected and is directed to a single output line. The QCA implementation of 8:1 multiplexer is shown in Figure 4.21(b)

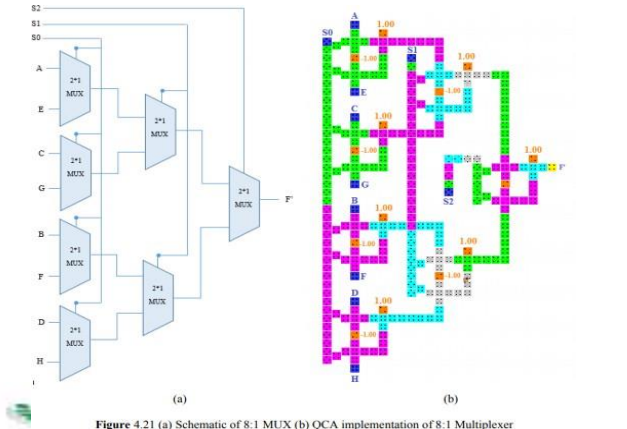


Figure 4.21 (a) Schematic of 8:1 MUX (b) QCA implementation of 8:1 Multiplexer

The comparison of various 8:1 multiplexers is given in Table 4.5

Structure	Parameters			
8: 1 Mux	**Cell Count**	**Cell Area (μm^2)**	**Total Area (μm^2)**	**Latency (Clock Phases)**
Mardiris et al. [107]	633	0.25	0.67	12
Vankamamadi et al. [169]	576	0.18	0.82	13
Proposed 8: 1 Mux	**288**	**0.093**	**0.44**	**7**

Table 4.5: Comparison of 4:1 Multiplexers

Thus a 2:1 multiplexer is used as the fundamental block in the design of higher order multiplexer circuits. Modular approach has been adopted in the designing of these circuits as it helps us to reduce the complexity as well as the latency.

Multiplexer as the Universal Structure

An electronic multiplexer allows a system to select one among the several input signals and forward it to the single output line. The signal to be forwarded to the output of the multiplexer is made by the selection lines. Thus the multiplexer is an extremely important part of signal control systems and this capability makes it possible to use it as a switch. All the logic gates can be designed with 2:1 multiplexer only. A multiplexer can be realized as the universal structure. The multiplexer already discussed can be further optimized so that the cell count can be further reduced. The QCA implementation and the simulation output of optimized 2:1 multiplexer are shown in Figure 4.23 & 4.24 respectively.

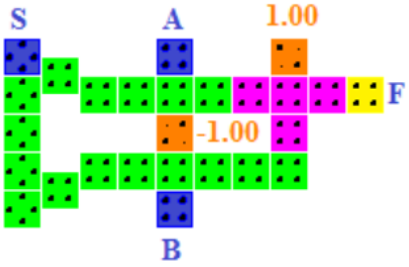


Figure 4.23. QCA implementation of 2:1 Multiplexer

The optimized design of 2:1 multiplexer only require 26 QCA cells for implementation. The delay from the input to output is only 2 clock phases. By appropriate selection of inputs and the selection line all the gates can be implemented by using the 2:1 multiplexer. Table 4.5 shows that by selecting the appropriate inputs (either by fixed polarity or variable polarity) the specific logic function/gate can be obtained.

IJFANS INTERNATIONAL JOURNAL OF FOOD AND NUTRITIONAL SCIENCES
ISSN PRINT 2319 1775 Online 2320 7876
 Journal Volume 10, Iss 11, 2021

Circuit	Selection Line S	Inputs		Output
		A	B	
2:1 Mux	S	A	B	$F = \overline{S}A + SB$
NOT Gate	S	1	0	$F = \overline{S}$
AND Gate	S	0	B	$F = S.B$
OR Gate	S	A	1	$F = S + B$
NAND Gate	S	0	B	$F = \overline{S.B}$
NOR Gate	S	A	0	$F = \overline{S + A}$
XOR Gate	S	A	$\overline{A}$	$F = \overline{S}.A + S.\overline{A}$
XNOR Gate	$\overline{S}$	A	$\overline{A}$	$F = \overline{S}.\overline{A} + S.A$

Table 4.6: Logic functions obtained using Multiplexer

A NOT gate function is obtained by fixing the polarity of input A as '1' and that of B as '0'. In this case, the output will be the compliment of S. In this case, S acts as input signal and A and B are used for their control. Similarly, the other logic functions have been obtained. The QCA implementation of the logic functions as given in the Table 4.6, are shown in the Figure 4.25(a-g)

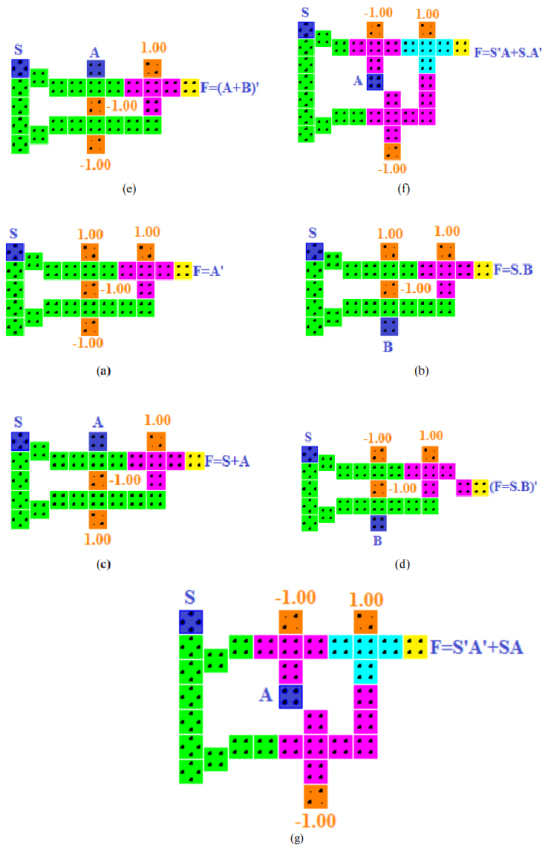


Figure 4.25. QCA implementation of MUX based (a) NOT gate (b) AND gate (c) OR gate (d) NAND gate (e) NOR gate (f) XOR gate (g) XNOR gate

Efficient Design of Digital Encoder using Quantum dot Cellular Automata

Digital computers operate on binary system, however we work on decimal numbers and the alphabets. Hence it is required to transmit the binary code for every stroke of alphanumeric keyboard. An encoder is a circuit which converts the alphanumeric characters to binary codes. An encoder can be classified as: i. Decimal to Binary Encoder ii. Octal to Binary Encoder iii. Hexadecimal to Binary Encoder iv. Octal to BCD Encoder Various implementations of encoders are reported in the literature [170-172]

Octal to Binary Encoder

An Octal to Binary Encoder has 8-input lines and 3-output lines. Corresponding to eight input octal numbers there exists three binary outputs. Consider an octal to binary encoder consisting of eight inputs labelled as D0, D1, D2, D3, D4, D5, D6 and D7. The three-bit binary output can be represented as A0, A1, A2 respectively. The block diagram and the circuit schematic of octal to binary encoder are shown in Figure 4.27

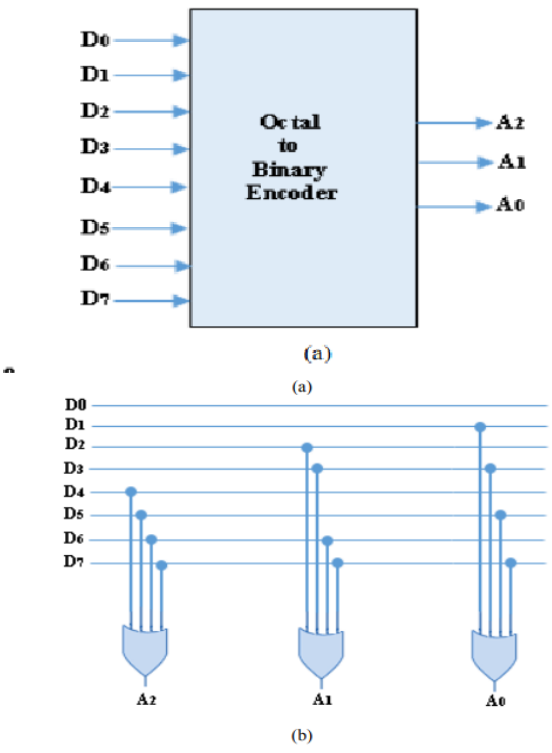


Figure:. (a) Block representation of encoder (b) Circuit schematic of octal to binary encoder.

The truth table of octal to binary encoder is shown in Table .

Inputs						Outputs				
D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7	A_2	A_1	A_0
1	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	0	1	1
0	0	0	0	1	0	0	0	1	0	0
0	0	0	0	0	1	0	0	1	0	1
0	0	0	0	0	0	1	0	1	1	0
0	0	0	0	0	0	0	1	1	1	1

Table 4.7: Truth table of Octal to Binary Encoder

D0 is not present in any of the equations since D0 is not connected to any gate. From the logic functions 1, 2 and 3, the input D7 is common to all the three outputs. Similarly, D3 is common to A0 and A1, D5 is common to A0 and A2, D6 is common to A1 and A2 respectively. Remaining all the inputs are applied individual. The QCA design of an octal to binary encoder is depicted in Figure 4.28.

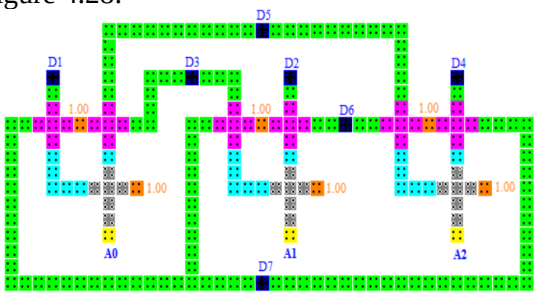


Figure 4.28. QCA design of Octal to Binary Encoder

The proposed design of octal to binary encoder is efficient in term of cell count, cell area, total area, latency and the complexity as compared to already reported designs. The proposed design of the octal to binary encoder has a cell count of 201 cells with a cell area of 0.06512 µm2 and total circuit area of 0.20930 µm2 , and the latency of 4 clock phases. Moreover, the proposed design doesn't require any crossover as it is implemented using the single layer design. The simulation outcome of an octal to binary encoder is shown in Figure.

Code Converters

The availability of a large variety of codes for the same discrete elements of information results in the use of different codes by different digital systems. Sometimes it is necessary to use the output of one system as the input to the other. Thus a conversion circuit must be inserted between the two systems if each uses different codes for the same information. A code converter is a circuit that makes the two systems compatible even though each uses the different codes. Indeed, code converters have proven to be so effective that the National Security Agency (NSA) has made a career out of creating and breaking codes. Code converters are used for protecting private information from spies. They are also used to enhance data portability and tractability. Code converters have also found applications in algorithm generation and communication [173-178]. Some of the major codes are as follows: Binary Code: A symbolic representation of data/information is called code. The base or radix of the binary number is 2. Hence, it has two independent symbols. The symbols used are 0 and 1. A binary digit is called a bit. A binary number consists of sequence of bits, each of which is either a 0 or 1. Each bit carries a weight based on its position relative to the binary point. The weight of each bit position is one power of 2 greater than the weight of the position to its immediate right. e. g. of binary number is 100011 which is equivalent to decimal number 35.

BCD Code: Numeric codes used to represent decimal digits are called Binary Coded Decimal (BCD) codes. A BCD code is one, in which the digits of a decimal number are encoded-one at a time into group of four binary digits. There are a large number of BCD codes in order to represent decimal digits 0, 1, 2,9, it is necessary to use a sequence of at least four binary digits. Such a sequence of binary digits which represents a decimal digit is called code word. Gray Code: It is a non-weighted code; therefore, it is not suitable for arithmetic operations. It is a cyclic code because successive code words in this code differ in one-bit position only i.e. it is a unit distance code. Excess-3 Code: It is a non-weighted code. It is also a self-complementing BCD code used in decimal arithmetic units. The Excess-3 code for the decimal number is performed in the same manner as BCD except that decimal number 3 is added to each decimal unit before encoding it to binary. Besides these codes there are large number of codes available. Various implementations are reported [179-181] using QCA. The implementation of code converters in QCA needs their representation at conventional gate level first. For getting these

ISSN PRINT 2319 1775 Online 2320 7876
 Journal Volume 10, Iss 11, 2021

representations, the conversion tables are depicted in Table 4.8 and Table 4.9 respectively.

Binary –Gray Code Conversion	
I (ABCD)	O (WXYZ
0000	0000
0001	0001
0010	0011
0011	0010
0100	0110
0101	0111
0110	0101
0111	0100
1000	1100
1001	1101
1010	1111
1011	1110
1100	1010
1101	1011
1110	1001
1111	1000

Table 4.8: Binary to Gray Code Conversion

2421-BCD Code conversion		5421-BCD Code conversion		BCD-84-2-1 Code conversion		BCD -2421-Code conversion		BCD-5421 Code conversion		BCD-Ex-3 Code conversion		BCD-Gray Code Conversion		Ex-3-BCD Code conversion	
A(ABCD)	*Z(WXYZ)*	*A(ABCD)*	*Z(WXYZ)*	*A(ABCD)*	*Z(WXYZ)*	*A(ABCD)*	*Z(WXYZ)*	*A(ABCD)*	*Z(WXYZ)*	*A(ABCD)*	*Z(WXYZ)*	*A(ABCD)*	*Z(WXYZ)*	*A(ABCD)*	*Z(WXYZ)*
0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0011	0000	0000	0011	0000
0001	0001	0001	0001	0001	0111	0001	0001	0001	0001	0001	0100	0001	0001	0100	0001
0010	0010	0010	0010	0010	0110	0010	0010	0010	0010	0010	0101	0010	0011	0101	0010
0011	0011	0011	0011	0011	0101	0011	0011	0011	0011	0011	0110	0011	0010	0110	0011
0100	0100	0100	0100	0100	0100	0100	0100	0100	0100	0100	0111	0100	0110	0111	0100
1011	0101	0101	0101	0101	1011	0101	1011	0101	0101	0101	1000	0101	0111	1000	0101
1100	0110	0110	0110	0110	1010	0110	1100	0110	0110	0110	1001	0110	0101	1001	0110
1101	0111	0111	0111	0111	1001	0111	1101	0111	0111	0111	1010	0111	0100	1010	0111
1110	1000	1011	1000	1000	1000	1000	1110	1000	1011	1000	1011	1000	1100	1011	1000
1111	1001	1100	1001	1001	1111	1001	1111	1001	1100	1001	1100	1001	1101	1100	1001

Table 4.9: Conversion of Various types of codes.

The logic expressions between the input and output vectors of the various code converter circuits mentioned in the Tables 4.8 and 4.9 are given in Table 4.10.

Circuit	Outputs			
	W	X	Y	Z
Binary-Gray Code Converter	A	$A \oplus B$	$B \oplus C$	$C \oplus D$
2421-BCD Code Converter	BC	$B\overline{C} + A\overline{B}$	$A \oplus C$	D
5421-BCD Code Converter	A	$\overline{A}B$	$\overline{A}C$	$\overline{A}D + AB$
BCD - 84-2-1 Code Converter	$A + BC + BD$	$\overline{B}D + \overline{B}C + B\overline{C}D$	$C \oplus D$	D
BCD - 2421 Code Converter	$A + BC + BD$	$A + BC + B\overline{D}$	$A + \overline{B}C + B\overline{C}D$	D
BCD - 5421 Code Converter	A	$B + AD$	$C + A\overline{D}$	$A \oplus D$
BCD – EX-3 Code Converter	$A + BC + BD$	$\overline{B}D + \overline{B}C + B\overline{C}D$	$\overline{C \oplus D}$	$\overline{D}$
BCD – Gray Code Converter	A	$A \oplus B$	$B \oplus C$	$C \oplus D$
EX-3 – BCD Code Converter	$AB + ACD$	$\overline{B}C + \overline{B}D + BCD$	$C \oplus D$	$\overline{D}$

Table 4.10: Relationship between the input and output vectors of various Code Converters.

V. SIMULATION RESULTS

DESIGN SUMMARY

Device Utilization Summary (estimated values)			
Logic Utilization	Used	Available	Utilization
Number of Slice LUTs	4982	204000	2%
Number of fully used LUT-FF pairs	0	4982	0%
Number of bonded IOBs	131	600	21%

The above result represents the synthesis implementation by using the Xilinx ISE software. From the above table, it is observed that only 4982 look up tables are used out of available204000. It indicates very less area(2%) was used for the proposed design.

TIME SUMMARY

```
LUT2:I0->O              1   0.043   0.000  div1/Madd_GND_49_o_GND_49_o_a
MUXCY:S->O              1   0.230   0.000  div1/Madd_GND_49_o_GND_49_o_a
XORCY:CI->O             2   0.251   0.347  div1/Madd_GND_49_o_GND_49_o_a
LUT4:I2->O              1   0.043   0.000  div1/Msub_n0258_Madd_lut<30>
MUXCY:S->O              0   0.230   0.000  div1/Msub_n0258_Madd_cy<30> (
XORCY:CI->O             1   0.251   0.289  div1/Msub_n0258_Madd_xor<31>
LUT5:I4->O              1   0.043   0.279  Mmux_out110 (out_0_OBUF)
OBUF:I->O                   0.000          out_0_OBUF (out<0>)
----------------------------------------
Total                      54.238ns (31.895ns logic, 22.343ns route)
                                    (58.8% logic, 41.2% route)
```

The above result represents the time consumed such as path delays by using the Xilinx ISE software.the consumed path delay is 54.238ns.

POWER SUMMARY

The above result represents the power consumed by using the Xilinx ISE software. The consumed power is 0.143uw.

VI. CONCLUSION

In this paper, we proposes a novel design of QCA comparator using five-input majority gate. The proposed comparator is simulated by XILINX-ISE and simulation shows the logic function of the proposed comparator is correct. Compared with previous comparator design, the proposed comparator has the least delay.

REFERENCES

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

[2] M. T. Niemer and P. M. Kogge, "Problems in designing with QCAs Layout = Timing," *Int. J. Circuit Theory Appl.*, vol. 29, no. 1,pp. 49–62, 2001.

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

[4] W. Liu, L. Lu, M. O'Neill, and E. E. Swartzlander, Jr., "Design rules for quantum-dot cellular automata," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2011, pp. 2361–2364. [5] K. Kim, K. Wu, and R. Karri, "Toward designing robust QCA architectures in the presence of sneak noise paths," in *Proc. IEEE Design, Autom. Test Eur. Conf. Exhibit.*, Mar. 2005, pp. 1214–1219.

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

[7] K. Walus, G. A. Jullien, and V. S. Dimitrov, "Computer arithmetic structures for quantum cellular automata," in *Proc. Asilomar Conf.Sygnals, Syst. Comput.*, Nov. 2003, pp. 1435–1439.