

Designing and Implementation of Adder & Subtractor Circuits in Quantum dot Cellular Automata

Thesis submitted in partial fulfillment
of the requirements for the degree of

Master of Technology
in
Electronics and Communication Engineering

by

Shobit Agarwal
14MEC013
shobit.agrawal27@gmail.com

Under Guidance of
Dr. Akshaykumar Salimath



Department of Electronics and Communication Engineering
The LNM Institute of Information Technology, Jaipur

June 2016

Designing and Implementation of Adder & Subtractor Circuits in Quantum dot Cellular Automata

Thesis submitted in partial fulfillment
of the requirements for the degree of

Master of Technology
in
Electronics and Communication Engineering

by

Shobit Agarwal
14MEC013
shobit.agrawal27@gmail.com

Under Guidance of
Dr. Akshaykumar Salimath



Department of Electronics and Communication Engineering
The LNM Institute of Information Technology, Jaipur

June 2016

Copyright © SHOBIT AGARWAL, 2016

All Rights Reserved

This document is copyrighted material. Under copyright law, no parts of this document may
be reproduced without expressed permission of the author.

The LNM Institute of Information Technology
Jaipur, India

CERTIFICATE

This is to certify that the thesis entitled "*Designing and Implementation of Adder & Subtractor Circuits in Quantum dot Cellular Automata*" submitted by *Shobit Agarwal (14MEC013)* towards partial fulfillment of the requirements for the degree of Master of Technology (M. Tech), is a bonafide record of work carried out by him/her at the Department of Electronics and Communication Engineering, The LNM Institute of Information Technology, Jaipur, (Rajasthan) India, during the academic session 2015-2016 under my/our supervision and guidance. the content of the thesis is original and have not been submitted elsewhere for award of any other degree. In my/our opinion, this thesis is of the standard required for the award of the said degree.

Date

Adviser: Dr. Akshaykumar Salimath

To Backbone of my Life
A Guide, An Inspiration, My Elder Brother
"Gaurav Mittal"

Acknowledgments

First of all, I would like to thank Dr. Akshaykumar Salimath for providing me an opportunity to work under his kind leadership and focussed guidance. I could have not completed this work without your guidance. He has a great knowledge in this field and always ready for discussion about problems and their solutions. I am truly indebted to him for encouraging the same pursuit and enthralments in me and, of course his direction and his assistance during the years as his student. I would like to thank my parents, brothers and sister for giving me so much love, encouragement and standing in my odd and even times. I am whole heartedly thankful to my parents who always have shown believe in me and allowed me to pursue Post Graduation. I do not have words to adequately describe my deep gratitude for all they have provided me; I hope to show them my appreciation in the forthcoming years. I would also like to thank Jatan Agarwal who has been a good friend and guardian away from my home. I would also thank all my friends Shivam, Anshul, Venkata Rajasekhar and Priya Gupta who stood by me all the time and being helpful whenever needed.

Abstract

As per Moore's law of 1975, the number of devices integrated over a unit area of a single chip doubles every 18 months. Nowadays, most of the devices are being fabricated using CMOS technology. Though these devices are providing you higher density as well as speed but are lacking in Power dissipation and other losses ex. Leakage current. As per scenario of the year 2015, we have reached up to technology node of 22nm. At this level, we are somehow managing these parameters into significant ranges. As we go to further lower level, those losses become significant and hence devices are not efficient. Due to this, researchers are looking for other alternate technologies which can replace CMOS. Quantum-dot Cellular Automata (QCA) is one of the novel nanotechnologies that is being considered as a possible replacement for CMOS. This technology works on the principle of coulomb repulsion and hence there is no leakage current. This thesis firstly covers designing of various combinational and sequential circuits along with logic gates. Later on, some modifications are proposed in terms of area, power dissipation, and speed. At last one Adder/ Subtractor circuit is designed and verified with QCADesigner tool 2.0.3.

Contents

Chapter	Page
1 Introduction	1
1.1 The Area of Research	1
1.2 Motivation	2
1.3 Organization of Thesis	4
2 Literature Review	5
2.1 Background	5
2.1.1 QCA Cell	5
2.1.2 QCA Cell-Cell Function	6
2.1.3 QCA Wire	7
2.1.4 QCA Wire Fan-out	8
2.1.5 Clocking of QCA	9
2.1.6 Crossover in QCA	11
2.1.6.1 Co-planer Crossover	11
2.1.6.2 Multi-Layer Crossover	12
2.2 Logic Primitives in QCA	14
2.2.1 Majority Voter Gate	14
2.2.2 QCA Inverter	15
3 Design, Implementation and Simulation of Combinational Circuits	17
3.1 Logic Gates	17
3.1.1 AND Gate	17
3.1.2 OR Gate	19
3.1.3 NOT Gate	19
3.1.4 NAND Gate	20
3.1.5 NOR Gate	21
3.1.6 XOR Gate	22
3.1.7 XNOR Gate	24
3.2 Different Variants of XOR Gate	25
3.2.1 Design-1	25
3.2.2 Design-2	26
3.2.3 Design-3	27
3.2.4 Design-4	28
3.2.5 Design-5	29
3.2.6 Design-6	30

3.2.7	Design-7	31
3.2.8	Performance Analysis of XOR Designs	32
3.3	MUX Designing	33
3.3.1	MUX Design-1	34
3.3.2	MUX Design-2	35
3.3.3	MUX Design-3	36
3.3.4	MUX Design-4	37
3.3.5	MUX Design-5	38
3.3.6	Comparison of 2×1 MUX Designs	39
3.3.7	Designing of 4×1 MUX by using 2×1 MUX	39
3.4	Adder Designing	41
3.4.1	1-Bit Adders	41
3.4.1.1	Design-1	42
3.4.1.2	Design-2	43
3.4.1.3	Design-3 (Multi-Layer Design)	44
3.4.1.4	Design-4	45
3.4.1.5	Comparison of 1-bit Adders	46
3.4.2	2-Bit Adders	46
3.4.3	4-Bit Adder	48
3.5	Multiplier Designing	50
4	Designing and Implementation of Random Access Memory (RAM) in QCA	53
4.1	Introduction	53
4.2	Basic RAM Design	53
4.3	RAM Cell with Set and Reset Switches	57
4.4	Advance RAM Structure	59
5	Designing and Implementation of Reversible Gates in QCA	61
5.1	Introduction	61
5.2	Basic Terms of Reversible Logic	62
5.3	Reversible Logic Gates	62
5.3.1	CNOT Gate	62
5.3.2	FEYNMAN Gate	64
5.3.3	Double FEYNMAN Gate	65
5.3.4	TOFFOLI Gate	66
5.3.5	FREDKIN Gate	68
5.3.6	BJN Gate	69
5.3.7	MCL Gate	71
5.4	Performance Analysis of Reversible Gates	72
6	Proposed Designs and Their Implementation	73
6.1	Adder Designing	73
6.1.1	1-Bit Adder	73
6.1.2	1-Bit Adder Modified Design	75
6.1.3	Comparison Study	76
6.1.4	2-Bit Adder	76
6.1.5	Power Analysis of Adder Designs	78

6.2	Adder/Subtractor Designing	80
7	Power Analysis in QCA	87
7.1	Power dissipation in QCA	87
7.1.1	Modelling Scheme	87
7.1.2	QCA Pro Simulation Setup	89
7.2	Power Analysis of implemented circuits	89
7.2.1	Logic Gates	90
7.2.1.1	AND Gate	90
7.2.1.2	OR Gate	94
7.2.1.3	NAND Gate:	95
7.2.1.4	NOR Gate:	96
7.2.1.5	XOR Gate Design-1:	96
7.2.1.6	XOR Gate Design-2:	97
7.2.1.7	XOR Gate Design-3:	98
7.2.1.8	XOR Gate Design-4:	98
7.2.1.9	XOR Gate Design-5:	99
7.2.1.10	XOR Gate Design-6:	99
7.2.1.11	XOR Gate Design-7:	100
7.2.1.12	XOR Gate Design-8:	100
7.2.1.13	Performance Analysis of XOR Gate Designs:	101
7.2.2	MUX Designs	103
7.2.2.1	MUX Design-1	103
7.2.2.2	MUX Design-2	104
7.2.2.3	MUX Design-3	104
7.2.2.4	MUX Design-4	105
7.2.2.5	MUX Design-5	105
7.2.2.6	Performance Analysis of MUX Designs	106
7.2.3	Adder Designs	108
7.2.3.1	Design-1	108
7.2.3.2	Design-2	109
7.2.3.3	Design-3	110
7.2.3.4	Performance Analysis of Adder Desings	111
7.2.4	Reversible Gates	113
7.2.4.1	CNOT Gate	113
7.2.4.2	FEYNMAN Gate	114
7.2.4.3	Double FEYNMAN Gate	115
7.2.4.4	TOFFOLI Gate	116
7.2.4.5	FREDKIN Gate	118
7.2.4.6	BJN Gate	119
7.2.4.7	MCL Gate	121
7.2.4.8	Performance Analysis of Reversible Gates	122
8	Conclusions and Future Work	125
8.1	Scope of future work	125
	Bibliography	126

List of Figures

Figure		Page
1.1 Taxonomy for emerging research information processing devices		2
2.1 QCA Cell Polarization for 90^0 Cell		6
2.2 QCA Cell Polarization for 45^0 Cell		6
2.3 QCA Cell to Cell Polarized Function		7
2.4 QCA Wire		7
2.5 QCA Wire Inverter Chain		8
2.6 QCA Wire Fan-out		8
2.7 Clock zones of QCA Clocking		9
2.8 Interdot barriers in a clocking zone		10
2.9 Operation of a wire in QCA zone clocking		10
2.10 Co-Planer Crossover QCA Layout		11
2.11 Simulation Result		12
2.12 Multi-Layer Crossover		12
2.13 Multi-Layer Crossover QCA Layout		13
2.14 Multi-Layer Crossover Simulation Result		13
2.15 QCA Implementation of MV Gate		14
2.16 Simulation Result of MV Gate		15
2.17 QCA Implementation of simple and robust inverter		15
2.18 Simulation Result for inverter		16
3.1 Symbol for AND Gate		17
3.2 QCA Layout and Simulation Result for AND Gate		18
3.3 Symbol for OR Gate		19
3.4 QCA Layout and Simulation Result for OR Gate		19
3.5 Symbol for NOT Gate		20
3.6 Symbol for NAND Gate		20
3.7 QCA Layout for NAND Gate		20
3.8 Simulation Result for NAND Gate		21
3.9 Symbol for NOR Gate		21
3.10 QCA Layout for NOR Gate		21
3.11 Simulation Result for NOR Gate		22
3.12 Symbol for XOR Gate		22
3.13 QCA Layout and Simulation Result for XOR Gate		23
3.14 Symbol for XNOR Gate		24

3.15	QCA Layout for XNOR Gate	24
3.16	Simulation Result for XNOR Gate	25
3.17	QCA Layout for XOR Design 1	25
3.18	Simulation Result XOR Design 1	26
3.19	QCA Layout for XOR Design 2	26
3.20	Simulation Result XOR Design 2	27
3.21	QCA Layout for XOR Design 3	27
3.22	Simulation Result XOR Design 3	28
3.23	QCA Layout for XOR Design 4	28
3.24	Simulation Result XOR Design 4	29
3.25	QCA Layout for XOR Design 5	29
3.26	Simulation Result XOR Design 5	30
3.27	QCA Layout for XOR Design 6	30
3.28	Simulation Result XOR Design 6	31
3.29	QCA Layout for XOR Design 7	31
3.30	Simulation Result XOR Design 7	32
3.31	Coherence Vector Simulation Engine Parameters	33
3.32	Basic Symbol for 2×1 MUX	33
3.33	QCA Layout for MUX Design 1	34
3.34	Simulation Result MUX Design 1	34
3.35	QCA Layout for MUX Design 2	35
3.36	Simulation Result MUX Design 2	35
3.37	QCA Layout for MUX Design 3	36
3.38	Simulation Result MUX Design 3	36
3.39	QCA Layout for MUX Design 4	37
3.40	Simulation Result MUX Design 4	37
3.41	QCA Layout for MUX Design 5	38
3.42	Simulation Result MUX Design 5	38
3.43	Basic Symbol 4×1 MUX using 2×1 MUX	39
3.44	QCA Layout and Simulation Result for 4×1 MUX using 2×1 MUX	40
3.45	Circuit Diagram for 1-bit Adder	41
3.46	QCA Layout for Adder Design 1	42
3.47	Simulation Result Adder Design 1	42
3.48	QCA Layout for Adder Design 2	43
3.49	Simulation Result Adder Design 2	43
3.50	QCA Layout for Adder Design 3	44
3.51	Simulation Result Adder Design 3	44
3.52	QCA Layout for Adder Design 4	45
3.53	Simulation Result Adder Design 4	45
3.54	2-bit Adder basic symbol	46
3.55	QCA Layout and Simulation Result for 2-Bit adder	47
3.56	4-bit Adder basic symbol	48
3.57	QCA Layout for 4-bit adder	49
3.58	Simulation Result for 4-bit adder	49
3.59	Simulation vector for 4-bit adder	50
3.60	Schematic Circuit for Multiplier	51

3.61 QCA Layout for 4-bit adder	51
3.62 Simulation Vector for 4-bit adder	52
4.1 D Latch Circuit Diagram	53
4.2 D Latch Circuit Diagram	54
4.3 Schematic Diagram for BAsic RAM Cell	55
4.4 QCA Layout for BAsic RAM Cell	55
4.5 Simulation Result for Basic RAM Cell	56
4.6 Schematic Diagram for RAM Cell with Set and Reset Switches	57
4.7 QCA Layout for RAM Cell with Set and Reset Switches	58
4.8 Simulation Result for RAM Cell with Set and Reset Switches	58
4.9 Schematic Diagram for advance RAM Cell	59
4.10 QCA Layout for advance RAM Cell	60
4.11 Simulation Result for advance RAM Cell	60
5.1 Symbol for CNOT Gate	63
5.2 QCA Layout of CNOT Gate	63
5.3 Simulation Result of CNOT Gate	63
5.4 Symbol for FEYNMAN Gate	64
5.5 QCA Layout of FEYNMAN Gate	64
5.6 Simulation Result of FEYNMAN Gate	64
5.7 Symbol for Double FEYNMAN Gate	65
5.8 QCA Layout of Double FEYNMAN Gate	65
5.9 Simulation Result of Double FEYNMAN Gate	66
5.10 Symbol for TOFFOLI Gate	66
5.11 QCA Layout of TOFFOLI Gate	67
5.12 Simulation Result of TOFOLLI Gate	67
5.13 Symbol for FREDKIN Gate	68
5.14 QCA Layout of FREDKIN Gate	68
5.15 Simulation Result of FREDKIN Gate	69
5.16 Symbol for BJT Gate	69
5.17 QCA Layout of BJT Gate	70
5.18 Simulation Result of BJT Gate	70
5.19 Symbol for MCL Gate	71
5.20 QCA Layout of MCL Gate	71
5.21 Simulation Result of MCL Gate	72
6.1 QCA Layout of 1-Bit Adder	74
6.2 Simulation Result of 1-Bit Adder	74
6.3 Cell reduction techniques (a)Diameter Wire (b)Stair Inverter	75
6.4 QCA Layout of Modified 1-Bit Adder	75
6.5 Simulation Result of Modified 1-Bit Adder	76
6.6 QCA Layout of 2-Bit Adder	77
6.7 Simulation Result of 2-Bit Adder	77
6.8 Power dissipation map for 1-bit adder ($1, 0.5 E_k$)	78
6.9 Power dissipation map for 1-bit adder modified ($1, 0.5 E_k$)	79
6.10 Power dissipation map for 2-bit adder ($1, 0.5 E_k$)	79

6.11	QCA Layout of XOR Gate	80
6.12	Simulation Result of XOR Gate	81
6.13	Power dissipation map for XOR Gate (1, 0.5 E_k)	81
6.14	QCA Layout of 8-bit Controllable inverter	82
6.15	Simulation Vector of 8-bit Controllable inverter	83
6.16	Simulation Result of 8-bit Controllable inverter	83
6.17	Adder/Subtractor Algorithm proposed in [23]	84
6.18	QCA Layout of 8-bit Adder/Subtractor	85
6.19	Simulation Vector of 8-bit Adder/Subtractor	85
6.20	Simulation Result of 8-bit Adder/Subtractor	86
7.1	1-Bit Adder design in QCA (a)Cell Layout; (b)Equivalent Bayesian Model	88
7.2	Design flow for QCA Pro tool	89
7.3	QCA Pro tool in action	90
7.4	Power dissipation map for AND Gate (1, 0.5Ek)	91
7.5	Power dissipation map for AND Gate (1, 1.0Ek)	91
7.6	Power dissipation map for AND Gate (1, 1.5Ek)	92
7.7	Power dissipation map for AND Gate (2, 0.5Ek)	92
7.8	Power dissipation map for AND Gate (2, 1.0Ek)	93
7.9	Power dissipation map for AND Gate (2, 1.5Ek)	93
7.10	Power dissipation map for OR Gate (1, 0.5Ek)	94
7.11	Power dissipation map for NAND Gate (1, 0.5Ek)	95
7.12	Power dissipation map for NOR Gate (1, 0.5Ek)	96
7.13	Power dissipation map for XOR Gate Design-1 (1, 0.5Ek)	97
7.14	Power dissipation map for XOR Gate Design-2 (1, 0.5Ek)	97
7.15	Power dissipation map for XOR Gate Design-3 (1, 0.5Ek)	98
7.16	Power dissipation map for XOR Gate Design-4 (1, 0.5Ek)	98
7.17	Power dissipation map for XOR Gate Design-5 (1, 0.5Ek)	99
7.18	Power dissipation map for XOR Gate Design-6 (1, 0.5Ek)	99
7.19	Power dissipation map for XOR Gate Design-7 (1, 0.5Ek)	100
7.20	Power dissipation map for XOR Gate Design-8 (1, 0.5Ek)	100
7.21	Avg. Leakage Energy dissipation (mev) for XOR Gate designs at $T=1^0K$	101
7.22	Avg. Switching Energy dissipation (mev) for XOR Gate designs at $T=1^0K$	102
7.23	Avg. Energy dissipation in circuit (mev) for XOR Gate designs at $T=1^0K$	103
7.24	Power dissipation map for MUX Design-1 (1, 0.5Ek)	103
7.25	Power dissipation map for MUX Design-2 (1, 0.5Ek)	104
7.26	Power dissipation map for MUX Design-3 (1, 0.5Ek)	104
7.27	Power dissipation map for MUX Design-4 (1, 0.5Ek)	105
7.28	Power dissipation map for MUX Design-5 (1, 0.5Ek)	105
7.29	Avg. Leakage Energy dissipation (mev) for MUX designs at $T=1^0K$	106
7.30	Avg. Switching Energy dissipation (mev) for MUX designs at $T=1^0K$	107
7.31	Avg. Energy dissipation in circuit (mev) for MUX designs at $T=1^0K$	107
7.32	Power dissipation map for Carry of Design-1 (1, 0.5Ek)	108
7.33	Power dissipation map for SUM of Design-1 (1, 0.5Ek)	108
7.34	Power dissipation map for Carry of Design-2 (1, 0.5Ek)	109
7.35	Power dissipation map for SUM of Design-2 (1, 0.5Ek)	109

7.36 Power dissipation map for Carry of Design-3 (1, 0.5Ek)	110
7.37 Power dissipation map for SUM of Design-3 (1, 0.5Ek)	110
7.38 Avg. Leakage Energy dissipation (mev) for Adder designs at $T=1^0K$	111
7.39 Avg. Switching Energy dissipation (mev) for Adder designs at $T=1^0K$	112
7.40 Avg. Energy dissipation in circuit (mev) for Adder designs at $T=1^0K$	112
7.41 Power dissipation map for CNOT Gate $P(1, 0.5Ek)$	113
7.42 Power dissipation map for CNOT Gate $Q(1, 0.5Ek)$	113
7.43 Power dissipation map for FEYNMAN Gate $P(1, 0.5Ek)$	114
7.44 Power dissipation map for FEYNMAN Gate $Q(1, 0.5Ek)$	114
7.45 Power dissipation map for Double FEYNMAN Gate $P(1, 0.5Ek)$	115
7.46 Power dissipation map for Double FEYNMAN Gate $Q(1, 0.5Ek)$	115
7.47 Power dissipation map for Double FEYNMAN Gate $R(1, 0.5Ek)$	116
7.48 Power dissipation map for TOFFOLI Gate $P(1, 0.5Ek)$	116
7.49 Power dissipation map for TOFFOLI Gate $Q(1, 0.5Ek)$	117
7.50 Power dissipation map for TOFFOLI Gate $R(1, 0.5Ek)$	117
7.51 Power dissipation map for FREDKIN Gate $P(1, 0.5Ek)$	118
7.52 Power dissipation map for FREDKIN Gate $Q(1, 0.5Ek)$	118
7.53 Power dissipation map for FREDKIN Gate $R(1, 0.5Ek)$	119
7.54 Power dissipation map for BJT Gate $P(1, 0.5Ek)$	119
7.55 Power dissipation map for BJT Gate $Q(1, 0.5Ek)$	120
7.56 Power dissipation map for BJT Gate $R(1, 0.5Ek)$	120
7.57 Power dissipation map for MCL Gate $P(1, 0.5Ek)$	121
7.58 Power dissipation map for MCL Gate $Q(1, 0.5Ek)$	121
7.59 Power dissipation map for MCL Gate $R(1, 0.5Ek)$	122
7.60 Avg. Leakage Energy dissipation (mev) for Reversible Gates at $T=1^0K$	123
7.61 Avg. Switching Energy dissipation (mev) for Reversible Gates at $T=1^0K$	124
7.62 Avg. Energy dissipation in circuit (mev) for Reversible Gates at $T=1^0K$	124

List of Tables

Table	Page
2.1 Truth Table for MV Gate	14
3.1 Truth Table for AND Gate	17
3.2 Truth Table for OR Gate	19
3.3 Truth Table for NOT Gate	20
3.4 Truth Table for NAND Gate	20
3.5 Truth Table for NOR Gate	21
3.6 Truth Table for XOR Gate	22
3.7 Truth Table for XNOR Gate	24
3.8 XOR Designs Comparative Study	32
3.9 Truth Table for 2×1 MUX	34
3.10 Comparative Study of 2×1 MUX Desings	39
3.11 Comparative Study of 1-Bit Adders	46
3.12 Truth Table for 4-bit adder ($A=0111$ and $B=1111$)	48
3.13 Truth Table for 2-Bit Multiplier	50
4.1 Truth Table for D Latch	54
4.2 Truth Table for D Latch	55
4.3 Truth Table for RAM Cell	57
4.4 Truth Table for Advance RAM Circuit	59
5.1 Truth Table for CNOT Gate	63
5.2 Truth Table for FEYNMAN Gate	64
5.3 Truth Table for Double FEYNMAN Gate	65
5.4 Truth Table for TOFFOLI Gate	66
5.5 Truth Table for FREDKIN Gate	68
5.6 Truth Table for BJT Gate	69
5.7 Truth Table for MCL Gate	71
5.8 Performance analysis of Reversible Gates	72
6.1 Comparison of Adder Designs	76
6.2 Power dissipation values for XOR Gate	82
6.3 Comparative Study of adder/subtractor designs	86
7.1 Power dissipation values for AND gate	94
7.2 Power dissipation values for OR gate	95

7.3	Power dissipation values for NAND gate	95
7.4	Power dissipation values for NOR gate	96
7.5	Avg. Leakage Energy dissipation (mev) at $T=1^0K$	101
7.6	Avg. Switching Energy dissipation (mev) at $T=1^0K$	101
7.7	Avg. Energy dissipation in circuit (mev) at $T=1^0K$	102
7.8	Avg. Leakage Energy dissipation (mev) at $T=1^0K$	106
7.9	Avg. Switching Energy dissipation (mev) at $T=1^0K$	106
7.10	Avg. Energy dissipation in circuit (mev) at $T=1^0K$	106
7.11	Avg. Leakage Energy dissipation (mev) at $T=1^0K$	111
7.12	Avg. Switching Energy dissipation (mev) at $T=1^0K$	111
7.13	Avg. Energy dissipation in circuit (mev) at $T=1^0K$	111
7.14	Avg. Leakage Energy dissipation (mev) at $T=1^0K$	122
7.15	Avg. Switching Energy dissipation (mev) at $T=1^0K$	122
7.16	Avg. Energy dissipation in circuit (mev) at $T=1^0K$	123

Chapter 1

Introduction

1.1 The Area of Research

As per Moore's law of 1975, the number of devices integrated over unit area of a single chip doubles every 18 months. Following this law, over the decades, CMOS technology is fulfilling the requirement for high speed, low size and higher density VLSI systems. Though current technology node of CMOS is 22nm, and this reduction has given increment in speed and density but it also reducing efficiency of circuit. If we continue to downscaling technology node, it will result into larger leakage current, crosstalk, power dissipation and electron-migration. There are static and dynamic power dissipations in CMOS. Former is caused due to leakage current while latter one is caused due to parasitic capacitive loads present at interconnects. According to The International Technology Roadmap for Semiconductors (ITRS), CMOS technology will continue till 2020 and hence scientists around the world are trying hard to find an alternate technology for CMOS. Few emerging technologies which can be used as an alternate to CMOS are as follows:

- Quantum-dot Cellular Automata (QCA)
- Single Electron Transistors (SETs)
- Tunneling Phase Logic (TPL)
- Spin Transistors
- Carbon Nanotube Transistors (CNTs)
- Resonant Tunneling Devices (RTD)

Quantum Dot Cellular Automata (QCA) is one of the alternative technologies which was firstly proposed by Craig S. Lent and P. Douglas Tougaw in 1993 and firstly fabricated in 1997. In QCA there are four dots along with two free electrons which specifies the logic states. Since here no voltage levels

are used, leakage current problem is not present in QCA. In QCA circuits information is being carried by electron moment which is present inside every cell because of coulombic interaction. QCA can be considered as a Novel technology that offers efficient nanoscale designing.

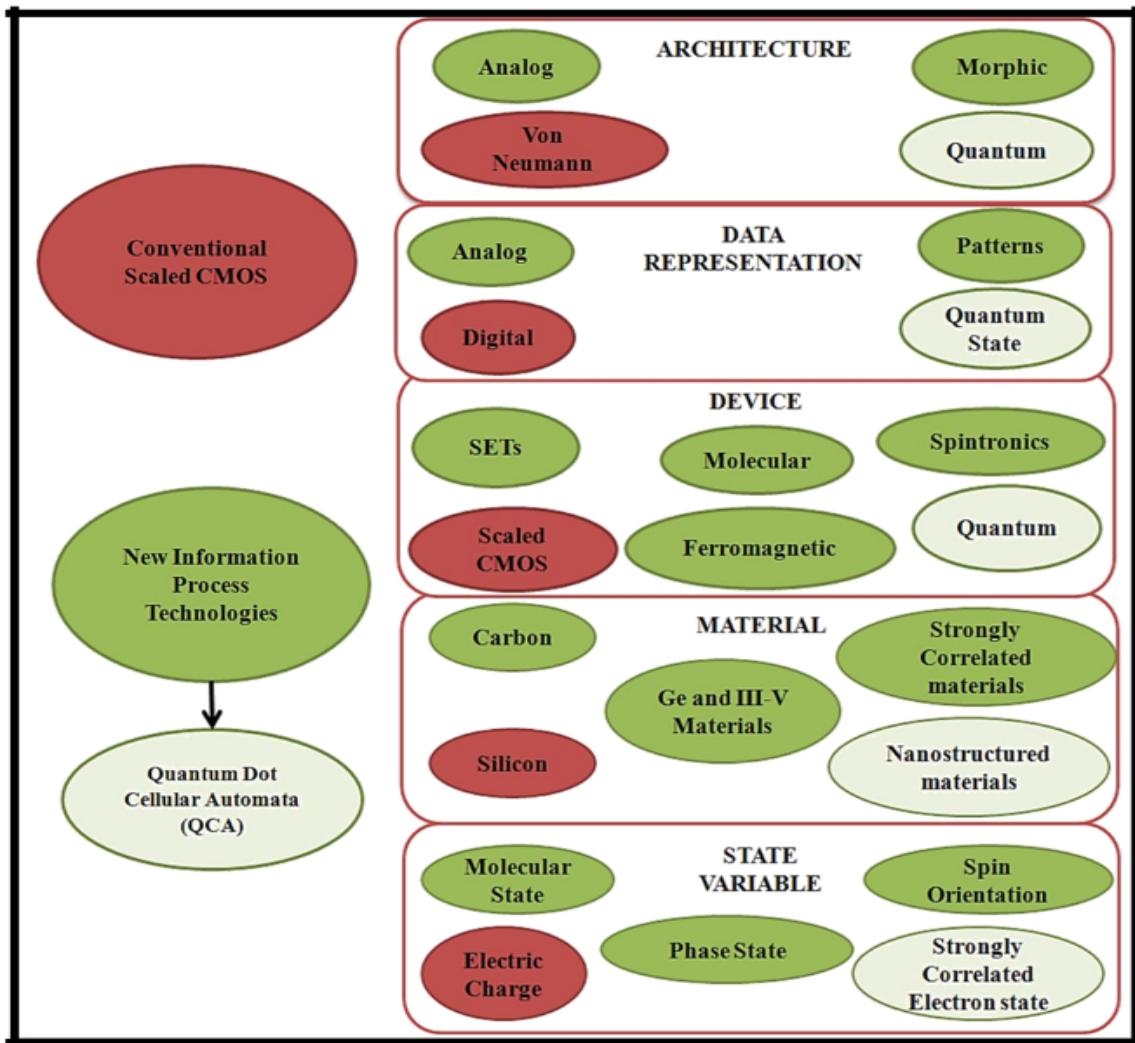


Figure 1.1: Taxonomy for emerging research information processing devices

1.2 Motivation

Since QCA is an emerging technology which provides high speed, high density with a very less power consumption as compared to CMOS technology. In past, different architectures have been implemented successfully on QCA as presented in [4] [7] [8] and more. In QCA technology all computations

are based on device to device coulomb interaction, There is no transfer of electrons at all, making it a potential option for very low power computing even below the traditional kT .

Another important aspect is power dissipation [5]. The chip temperature will see a rise because of Power dissipation and the device suffers, no matter whether it is ON or OFF. Number of intrinsic charge carriers n_i will increase, if the device is OFF. Equation as follows :

$$n_i = \exp^{\frac{-EG}{KT}}$$

And the majority carriers are less affected so the device becomes more intrinsic. Leakage current is directly proportional to minority carrier concentration, so the device might breakdown because leakage current will further increase the temperature.

When the device is ON, it will not be affected much by minority carriers, but V_T and μ decrease will lead to a change in I_D . Hence device might not meet required specifications.

Main reasons behind power dissipation in CMOS:

- Static power dissipation P_S
- Dynamic power dissipation P_D
- Short circuit power dissipation P_{SC}

Thus the total power dissipation, P_T is

$$P_T = P_S + P_D + P_{SC}$$

Static power dissipation due to:

- Sub-threshold conduction through OFF transistors
- Tunneling current through gate oxide
- Leakage through reverse biased diodes
- Contention current in rationed circuits

Dynamic power dissipation is due to Charging and discharging of load capacitances. Short circuit power dissipation will be there while both PMOS and NMOS are partially ON. While power dissipation in QCA is lower in comparison to CMOS. Further discussion about power dissipation in QCA has given a separate chapter.

The focus in this thesis is to design adders and adder/subtractor circuits using QCA designer tool. These designs are implemented successfully and simulated with QCADesigner tool 2.0.3. Another focus of this work is to carry out power dissipation analysis for different QCA circuits with the help of Bayesian network analysis.

1.3 Organization of Thesis

This thesis is organized in following manner.

Chapter 2: It consists of QCA history and basic QCA devices

Chapter 3: Consists of Basic Logic gates and other combinational circuits QCA implementation and their simulation results.

Chapter 4: Describes Random Access Memory (RAM) QCA implementation and their simulation results.

Chapter 5: Consists of Reversible Logic Gates QCA implementation and their simulation results.

Chapter 6: Shows proposed designs viz. 1-bit and 2-bit adders and an 8-bit adder/subtractor circuits along with their simulation results verified by QCADesigner tool 2.0.3

Chapter 7: Consists of power consumption analysis for all possible logic and sequential circuits which are compatible to QCADesigner tool 1.4.0

Chapter 8: Concludes the thesis work and discuss about future possibilities.

Chapter 2

Literature Review

2.1 Background

This was introduced in 1993 by Craig Lent, Quantum-dot Cellular Automata (QCA) is one of the emerging technologies likely to overtake the conventional CMOS technology [1] [2]. Quantum Dots are nanostructures created from semi-conductive material. A dot can be visualized as a quantum well. QCA uses arrays of coupled quantum dots to implement Boolean logic functions. QCA is the physical implementation of "classical" cellular automata by exploiting quantum mechanical phenomena. Conventional digital technologies use either voltages or currents to represent logic values whereas in QCA technology, position of the electrons represents the binary values [11]. The advantages of this technology are:

- High Operating Speed = Range of *Terahertz*
- Low Power Consumption = $100W/cm^2$
- High Device Density = $10^{12} devices/cm^2$

Before proceeding to the thesis, let's have a look on basic structures of QCA Technology which comprises of QCA cell, QCA wire, Majority Gate and Inverter. All these are required to develop various digital circuits at the nanotechnology level.

2.1.1 QCA Cell

A QCA Cell composed of four metal islands along with two electrons. Metal islands are known as Quantum dots and each quantum dot shows the quantum mechanical properties. They are located at cell corners. The square boundary is just for illustration only. Electrons of the cell are allowed to tunnel but restricted to go out of cell. Electrons always occupies antipodal sites as Coulomb repulsion is less in this case compared to when they are occupied adjacent dots. Hence they produce lowest energy state of

the system. If we consider a QCA cell then we can get two possible scenarios for electron occupancy. This is known as **Polarization of Cell**. Polarization (P) can be calculated as[4]:

$$P = \frac{(\rho_1 + \rho_3) - (\rho_2 + \rho_4)}{(\rho_1 + \rho_2 + \rho_3 + \rho_4)}$$

We have two types of QCA cell namely 90^0 or *NormalCell* and 45^0 or *RotatedCell*. The logic value “0” and “1” are represented as per Fig. 2.1 for 90^0 and Fig 2.2 shows for 45^0 cells. A QCA cell can get polarized by applying an external field or due to its neighbouring cell. When any of them is not present, *i.e.* electrons have not occupied any dot, State is known as *NULL State*.

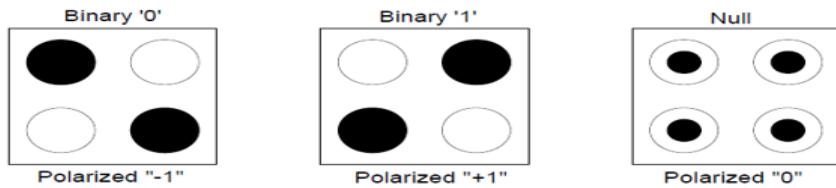


Figure 2.1: QCA Cell Polarization for 90^0 Cell

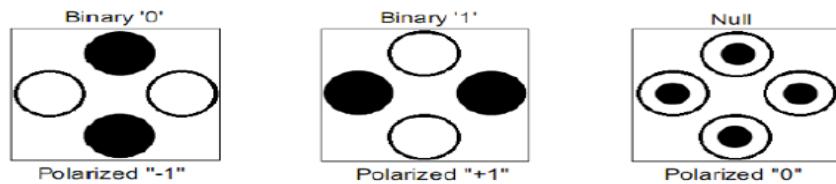


Figure 2.2: QCA Cell Polarization for 45^0 Cell

2.1.2 QCA Cell-Cell Function

The cell to cell response function for QCA Cell is shown in Figure 2.3. After observation we can conclude that the response is non-linear because weak polarization of one cell can result in strong polarization of neighbouring cell. It means that, in a wire, driving cell or any of the intermediate cells had a weak polarization, the follower cell may be strongly polarized. This can be understand by following example, Suppose we are transmitting logic 1 through a wire consisting finite number of cells. Hence it will take certain time to propagate. As we change our input to logic 0, then next cell would have been transmitting 1 and in next clock cycle it will take 0.

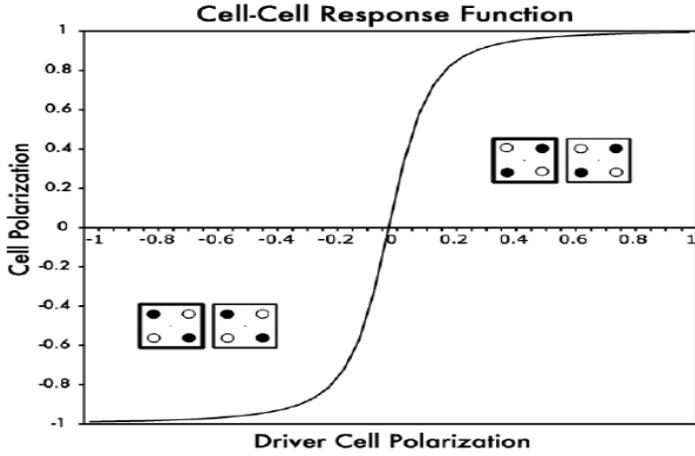


Figure 2.3: QCA Cell to Cell Polarized Function

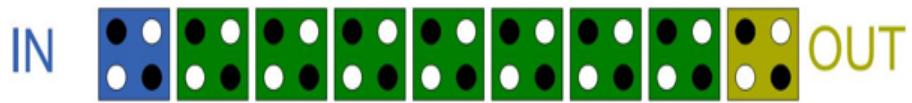
2.1.3 QCA Wire

If we put QCA cell one after another, QCA wire is formed. By the application of Coulombic interaction *i.e.* magnitude of electron repulsion from neighbouring cells we can find out the location of electrons inside QCA cell. Considering Fig. 2.4, we can observe certain colors as Blue, Green & Yellow. They are meant for Input, Output and normal cell under clock 0. As we apply logic 1 to input in wire of Fig. 2.4(a), leftmost green cell electrons occupies polarization of "+1" due to Coulombic interaction. This cycles repeats until the output is produced. Hence we can conclude that, A wire has all cell with same polarization.

Fig. 2.4(b) illustrates for logic 0.



(a) Transmission of Logic **1**



(b) Transmission of Logic **0**

Figure 2.4: QCA Wire

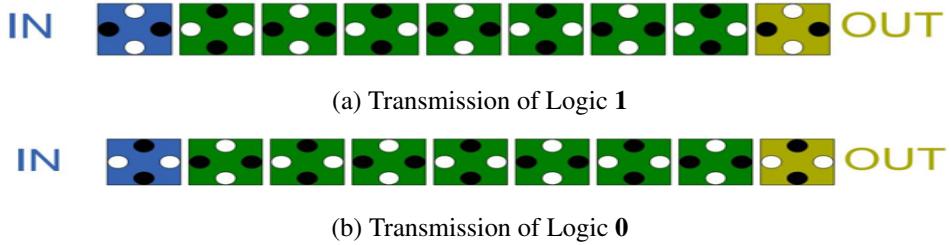


Figure 2.5: QCA Wire Inverter Chain

On the other side, When two cells of 45^0 are placed adjacent to each other, they always have opposite polarities, which results in the ground state of the system of those cells. The wire with 45^0 cells is called an *inverter-chain* because the logic level of any cell in the wire is the negation of its adjacent previous cell.

2.1.4 QCA Wire Fan-out

We can obtain the fan-out in QCA by using take away point concept of control system theory. Likewise, we can branch our main wire into required end points. These end-points then serve as inputs to other gates or final output. This is shown in Fig. 2.6(a) with input IN as logic 1. While designing complex circuits, we may need signal as well as its complement form simultaneously. There are several ways for it, Likewise: we can add another wire carrying complemented signal or put an inverter wherever complemented form is required. Here comes the role of Rotated cells in picture, as they can form inverter chain and can provide you signal and its compliment simultaneously by tapping off from correct cell pairs. AS we can see in Fig. 2.6(b), OUT_~IN is same as input while OUT_IN is inverted form.

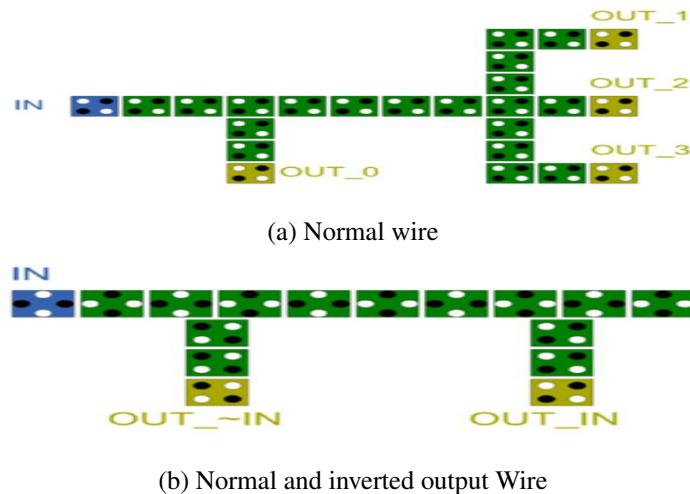


Figure 2.6: QCA Wire Fan-out

2.1.5 Clocking of QCA

This can be considered as the heart of QCA as without proper clocking it is not at all possible to design any circuit in QCA technology. There are two types of clocking used in QCA[21].

1. **Zone Clocking:** As implies its name, it has four zones of clock named as *Clock Zone 0*, *Clock Zone 1*, *Clock Zone 2*, *Clock Zone 3*. Each zone is 90^0 out of phase by its predecessor which implies Clock zone 1 is 90^0 ahead by zone 0. Each clock zone has four phases as *switch*, *hold*, *release* and *relax*. During *Switch* phase, an un-polarized cell becomes polarized and potential barrier raised. In *Hold* phase, barriers remain high and polarization is reserved. During *Release* phase, cell loses its polarization and become un-polarized again and lowers potential barrier. At last in *Relax* phase, barriers and cell remain low and un-polarized respectively. All those stages can be observed in Fig. 2.8.
2. **Continuous Clocking:** In this clocking scheme, we generate external electric field by some system of sub-merged electrodes.

Since, QCADesigner tool 2.0.3 supports only zone clocking thus we have used only zone clocking throughout work.

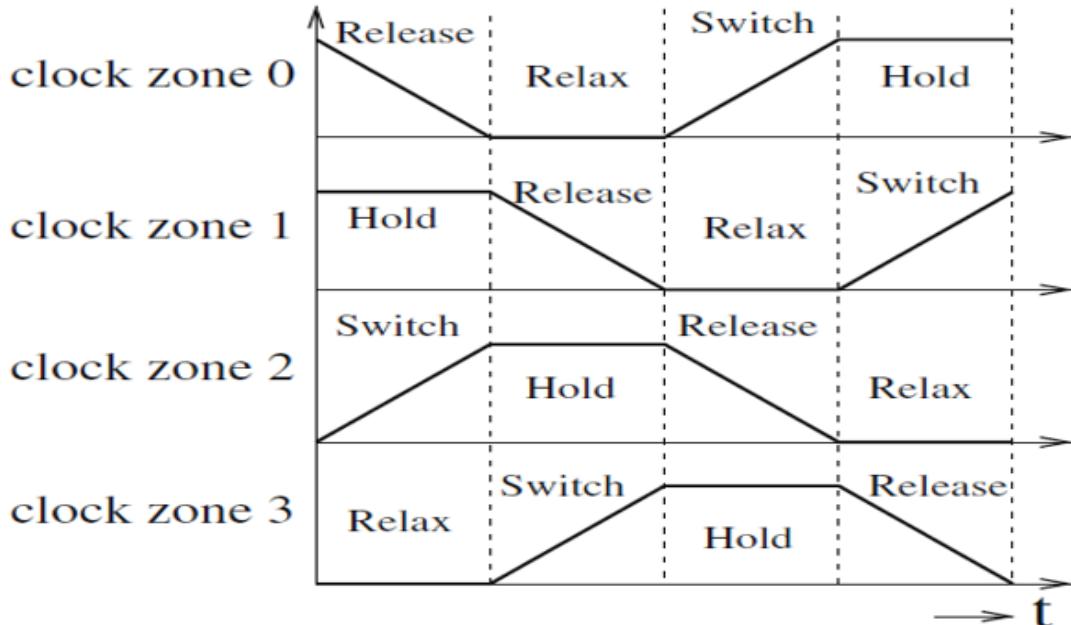


Figure 2.7: Clock zones of QCA Clocking

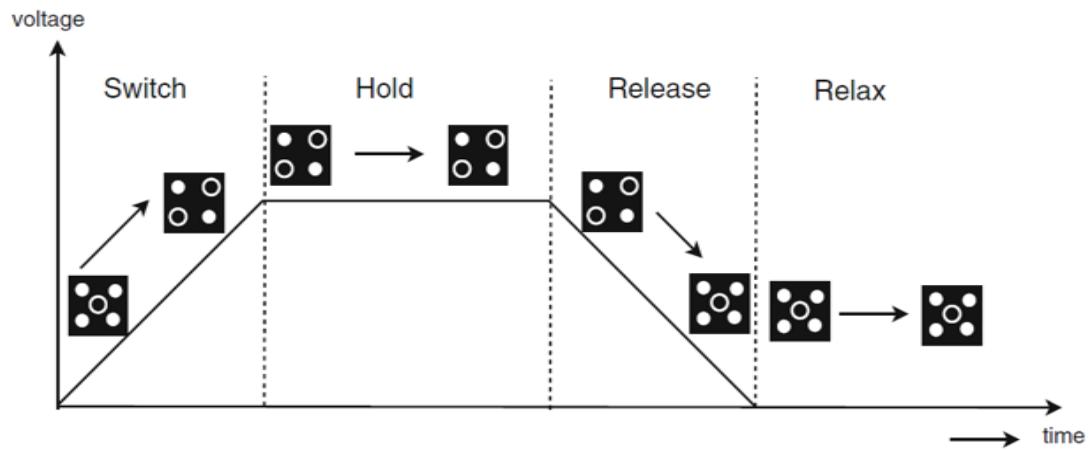


Figure 2.8: Interdot barriers in a clocking zone

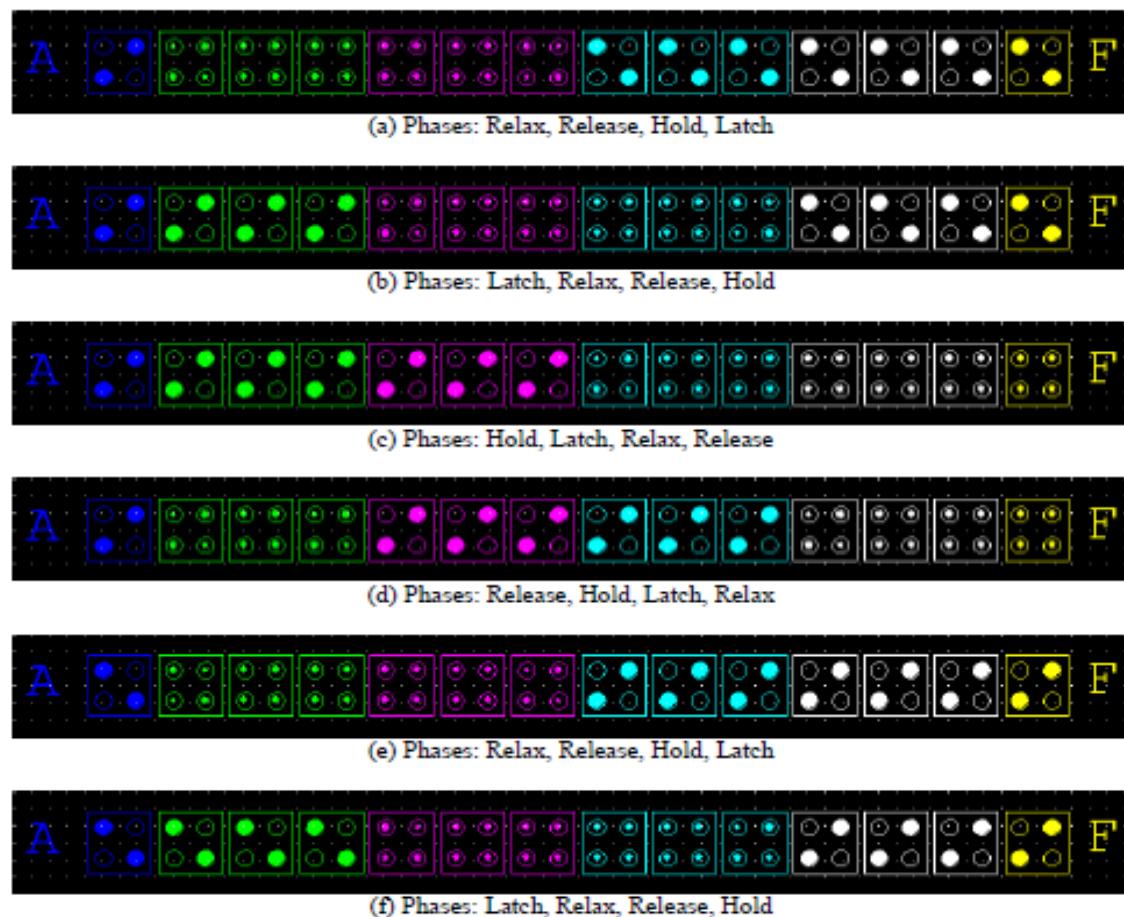


Figure 2.9: Operation of a wire in QCA zone clocking

2.1.6 Crossover in QCA

Crossover in QCA can be broadly classified as:

1. Co-Planer Crossover
2. Multi-Layer Crossover

2.1.6.1 Co-planer Crossover

This type of crossover is performed when circuit is planer *i.e.* both wires exist on same layer. As kink energy of normal cell and rotated cell, when used together, cancel out and results in zero, therefore if we design one wire with normal cells and another wire with rotated cells then they will not interact each other. The QCA layout and simulation result for Co-planer crossover is shown in fig. 2.10 and 2.11 respectively. As clear from fig. 2.11, A is propagated to OUT1 and B is propagated to OUT2 without interruption. Although, this is a good technique but also sensitive to fabrication issue. In order to resolve this problem, a descriptive analysis need to be carried out. Moreover, we must be very precise in placement of cells as mis-placement can result in form of crosstalk. Due to above said reasons, we generally avoid this crossover.

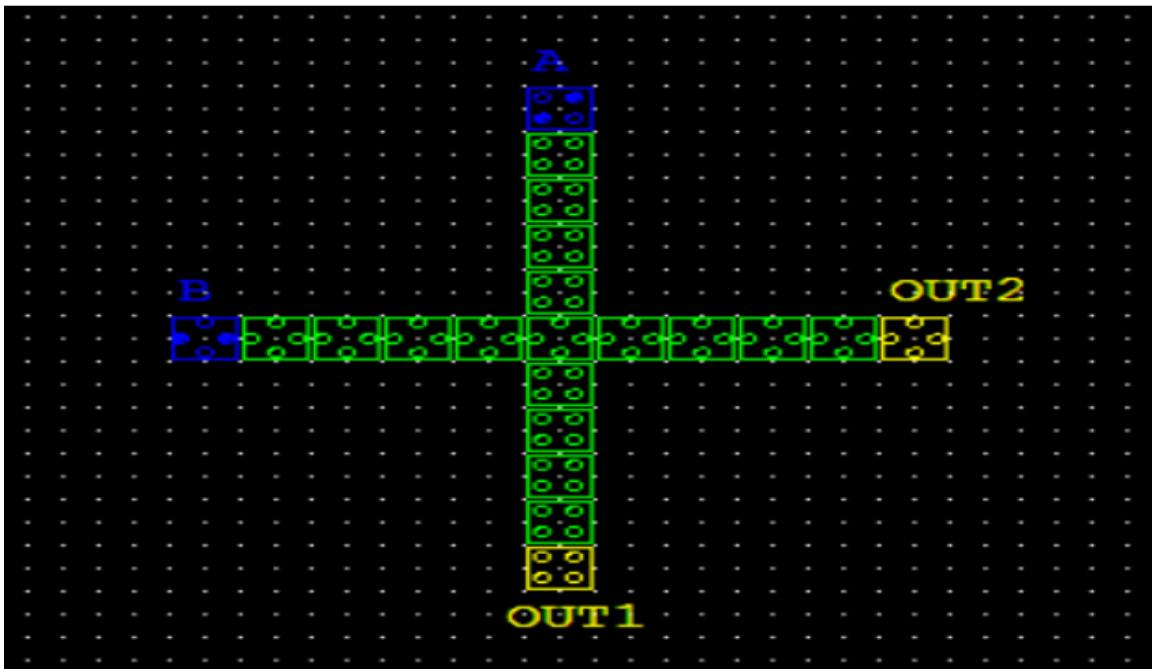


Figure 2.10: Co-Planer Crossover QCA Layout

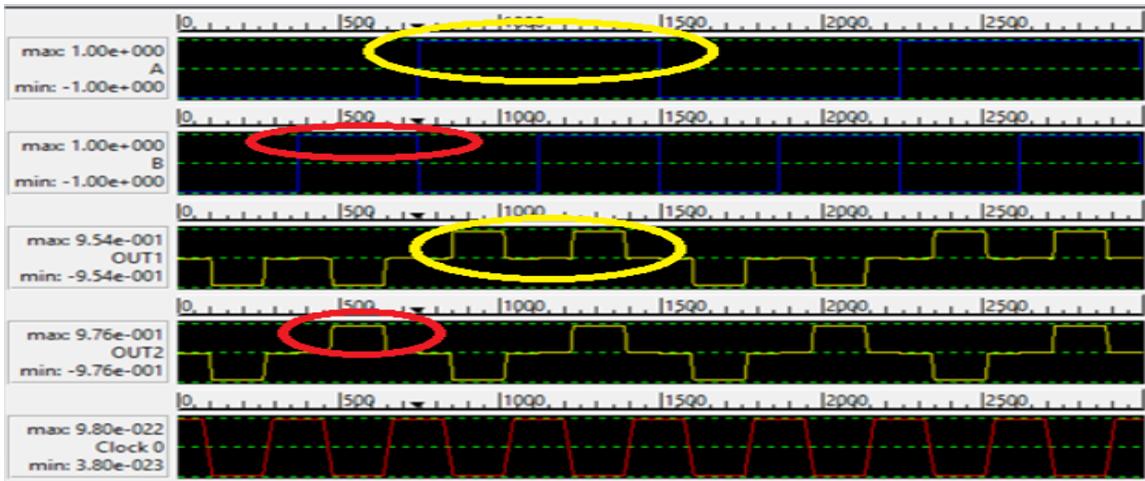


Figure 2.11: Simulation Result

2.1.6.2 Multi-Layer Crossover

Since Co-planer crossover is sensitive to various facts, here comes another methodology for crossover. Because as we start designing of complex circuits, it is almost impossible to avoid crossover completely. This works in following manner, we generally have 3 layers to form a crossover (Let name them as *Base*, *Via* and *Top*). Complete circuit, excluding crossover area, is designed on *Base* layer. then crossover signal travels through *Vertical cell* (having circle inside boundary) to *Top* layer and travels on *Top* layer through *Crossover Cell* (having cross structure) and finally returns to *Base* layer through *Vertical cell* again. Intermediate layer has only *Vertical cells*. This is very efficient way of crossover and also it solves co-planer crossover problems. In addition, it gives boost in area.

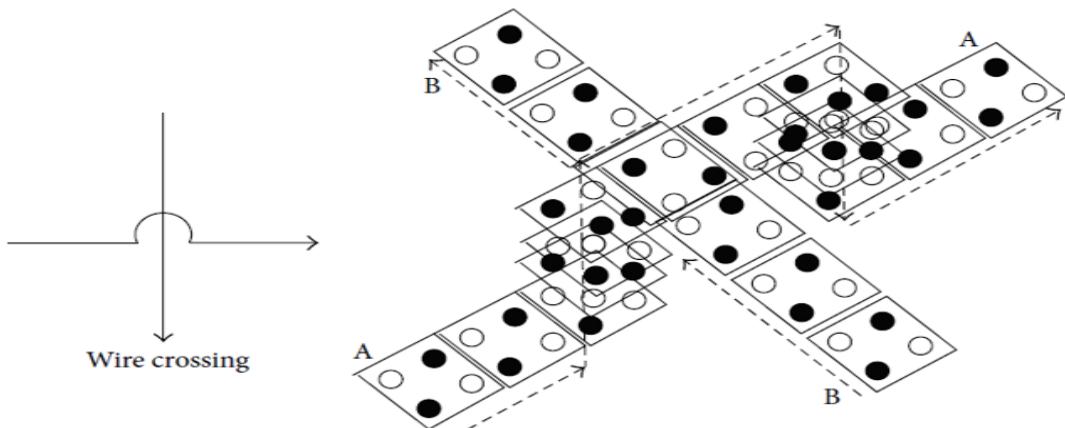


Figure 2.12: Multi-Layer Crossover

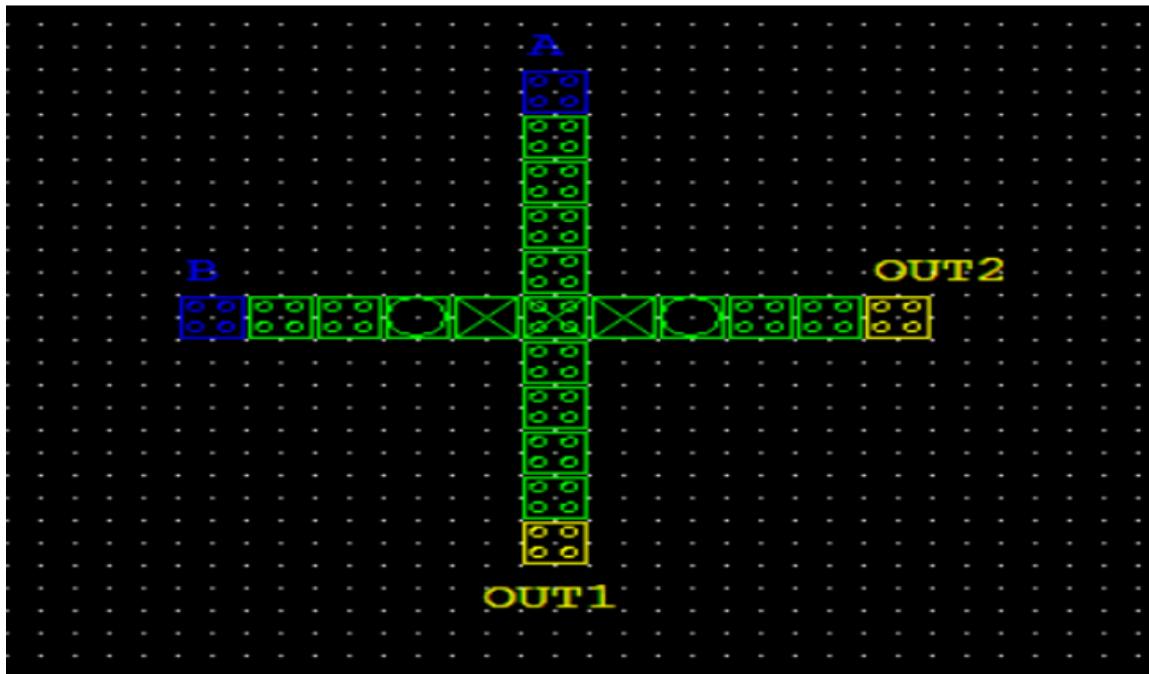


Figure 2.13: Multi-Layer Crossover QCA Layout

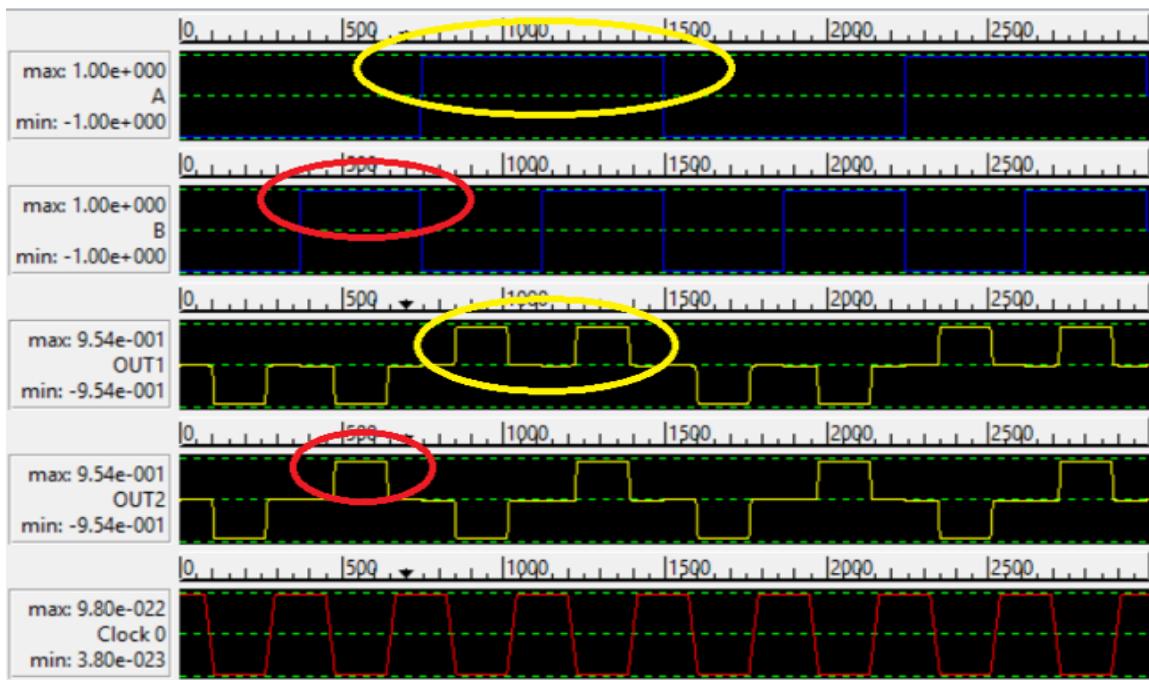


Figure 2.14: Multi-Layer Crossover Simulation Result

2.2 Logic Primitives in QCA

There are basically two logic primitives in QCA as follows[4]:

2.2.1 Majority Voter Gate

As its name implies, output state is same as majority of inputs state. Which means that if more inputs are “1” then output is “1” otherwise “0”. The truth table and logic function for MV gate is as below:

<i>A</i>	<i>B</i>	<i>C</i>	<i>Maj (A,B and C)</i>
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

Table 2.1: Truth Table for MV Gate

After solving K-Map for above table we obtain Boolean relation as:

$$M(A, B, C) = AB + BC + CA$$

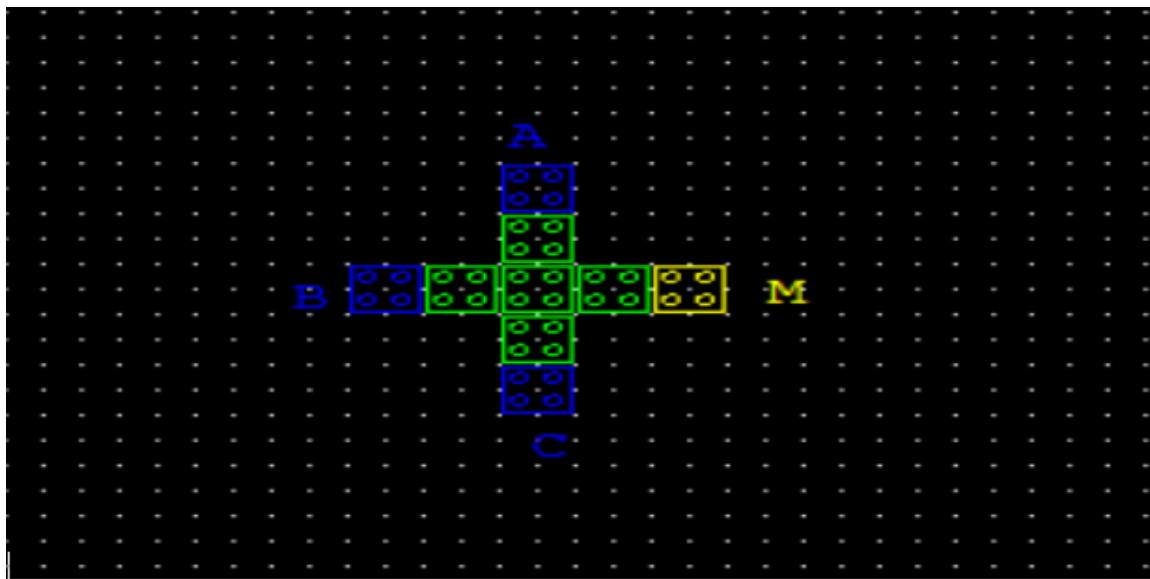


Figure 2.15: QCA Implementation of MV Gate

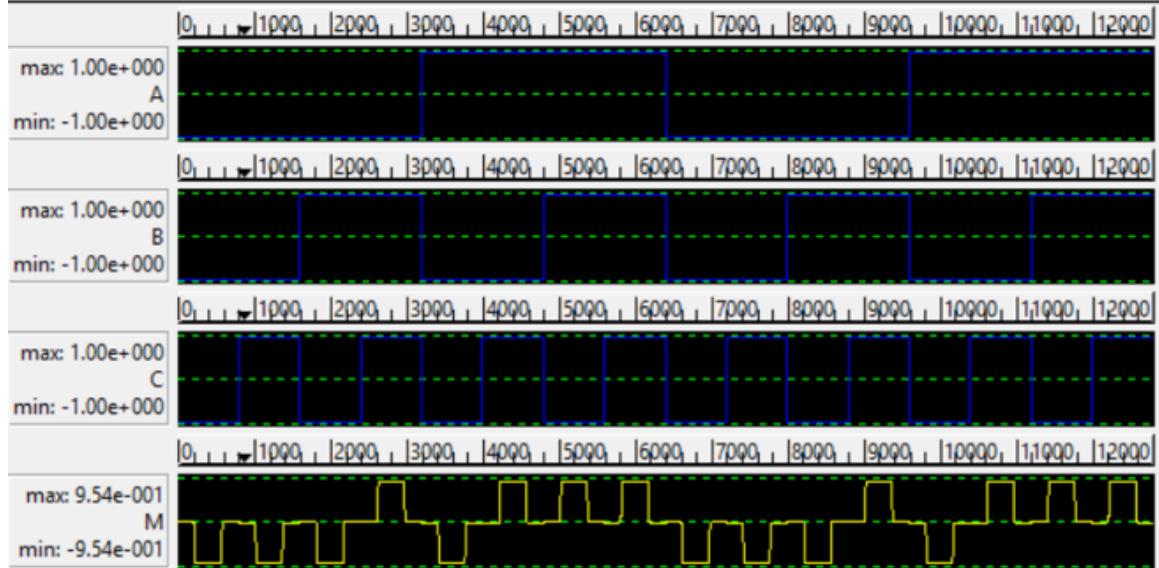


Figure 2.16: Simulation Result of MV Gate

2.2.2 QCA Inverter

QCA inverter is as same as NOT Gate. This is used to compute negation of input. In fig. 2.17 we can see two types of inverter. first one is obtained by displacing the cell w.r.t each other. This design is not efficient as displacement decreases polarization and hence electrons may stuck in metastable state. Alternative to this design, Robust Inverter, is shown in second figure. Although we get good results from this design but we have to tolerate # QCA Cells.

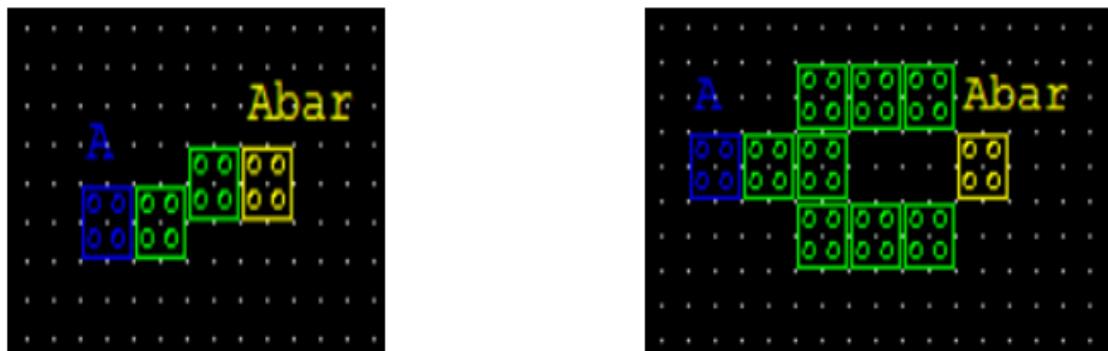


Figure 2.17: QCA Implementation of simple and robust inverter

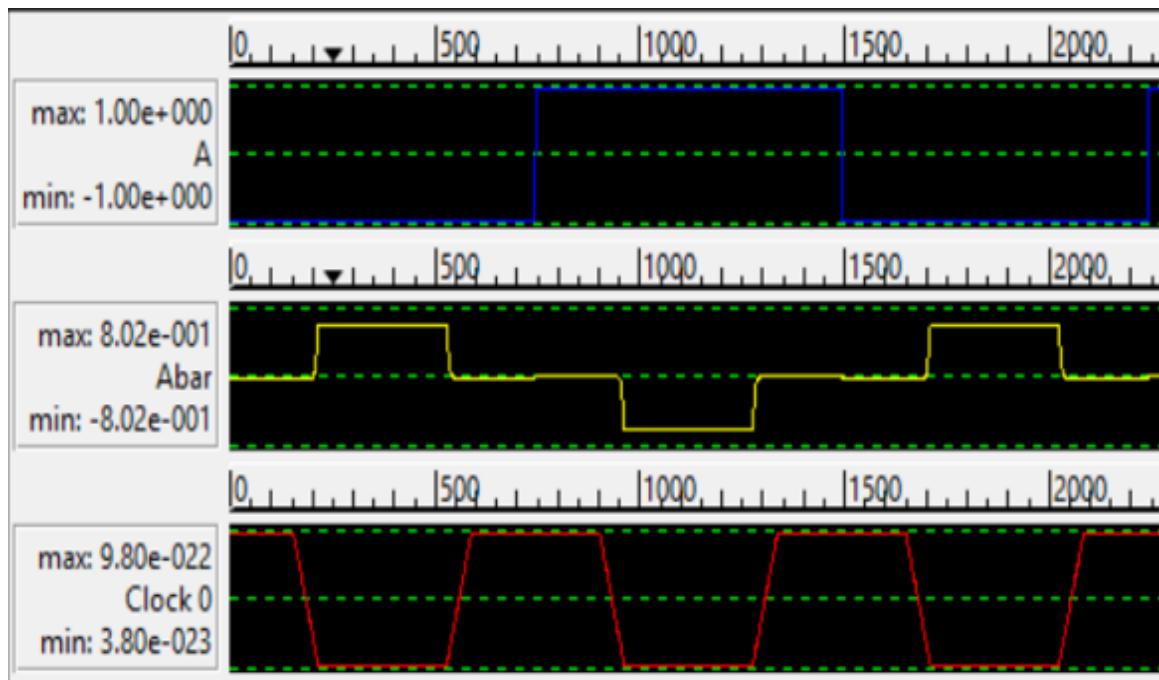


Figure 2.18: Simulation Result for inverter

Chapter 3

Design, Implementation and Simulation of Combinational Circuits

3.1 Logic Gates

Logic gates are the basic designs to implement any combinational circuit. It includes AND, OR, NOT, NAND, NOR, XOR, XNOR Gates[4].

3.1.1 AND Gate

Logic AND Gate may be considered as a series connection of switches which gives high output only when all of its input are at high level, otherwise output remains low.

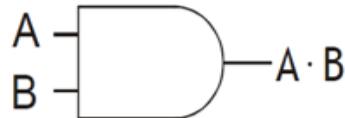
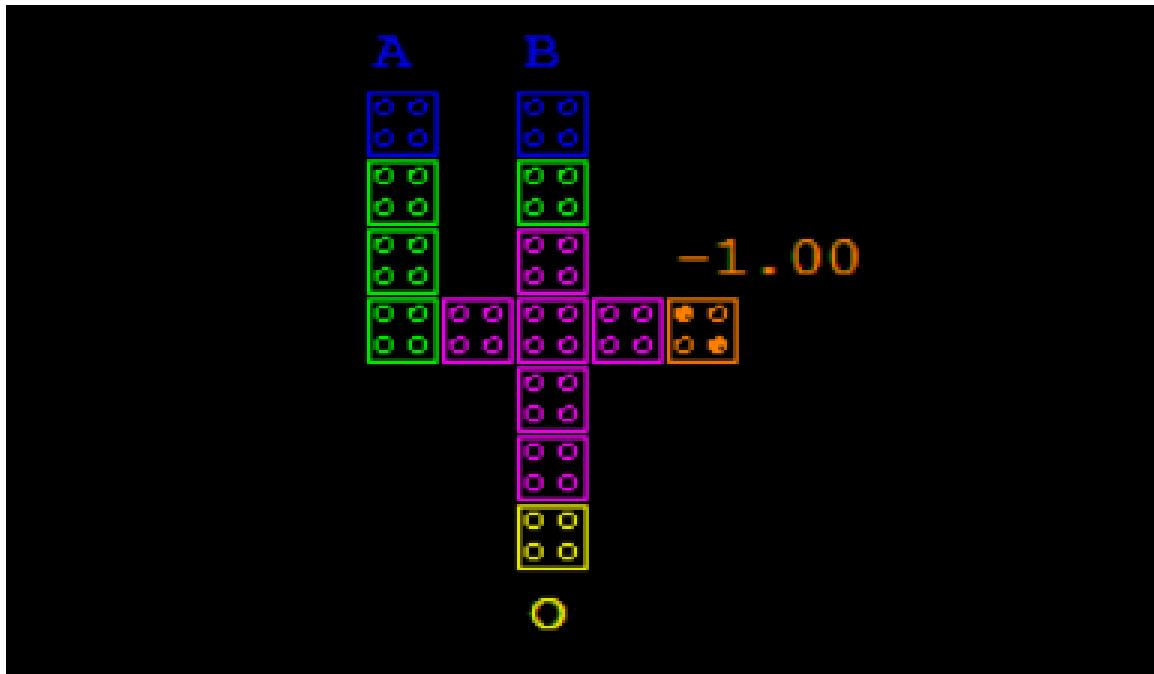


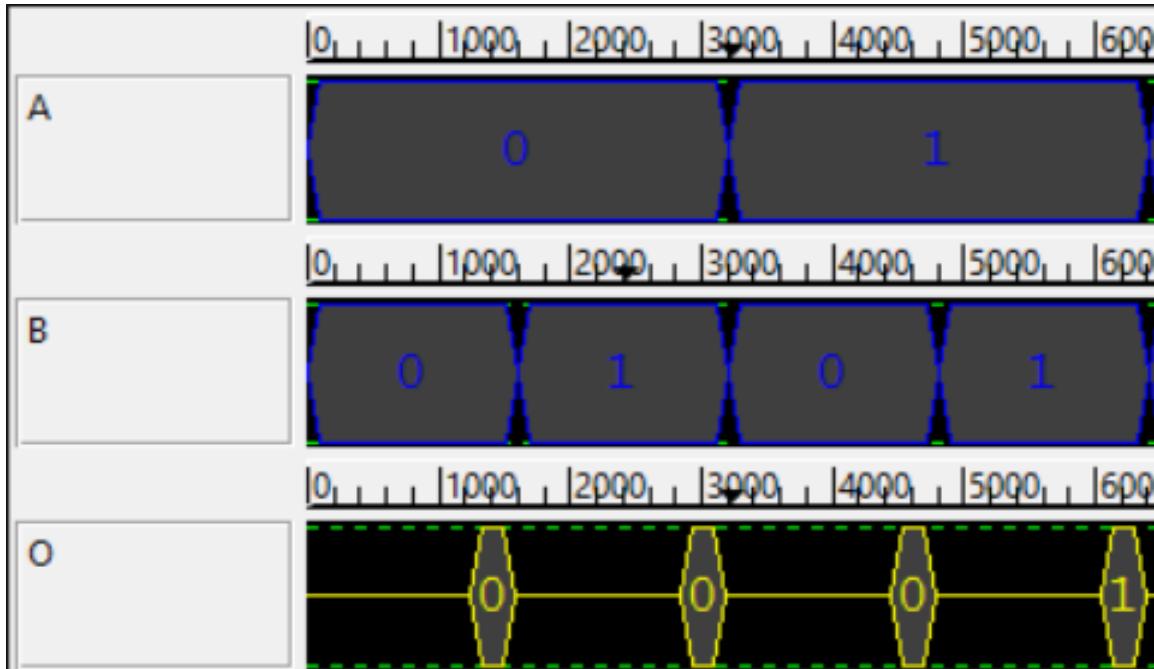
Figure 3.1: Symbol for AND Gate

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

Table 3.1: Truth Table for AND Gate



(a) QCA Layout for AND Gate



(b) Simulation Result for AND Gate

Figure 3.2: QCA Layout and Simulation Result for AND Gate

3.1.2 OR Gate

In similar fashion to AND gate, OR gate can also be realized using MV Gate by setting one of its input to “1”. OR Gate gives high output when any of its input is high.



Figure 3.3: Symbol for OR Gate

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

Table 3.2: Truth Table for OR Gate

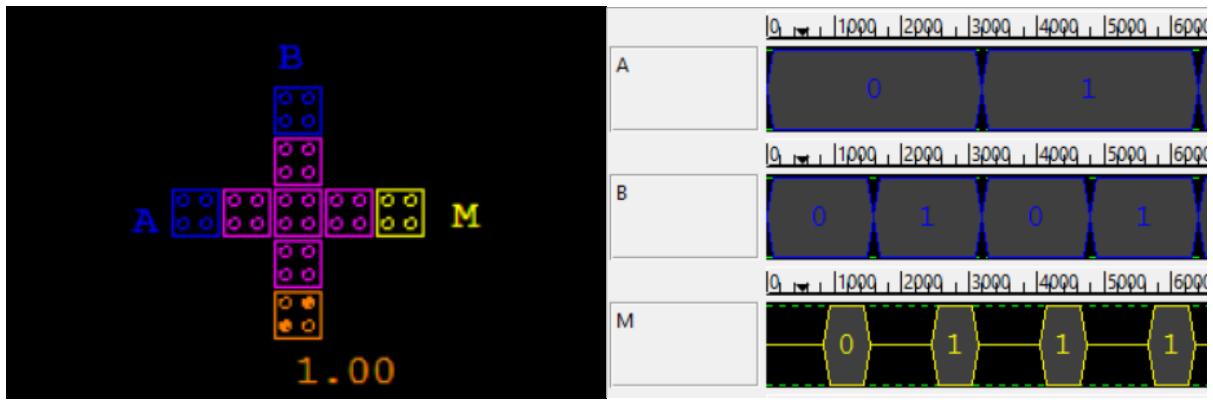
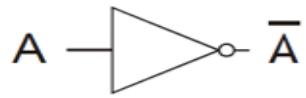


Figure 3.4: QCA Layout and Simulation Result for OR Gate

3.1.3 NOT Gate

This is used to produce complement of input. It is mostly used wherever we need signal and its complemented form. QCA inverter design directly works as NOT Gate which is described earlier in chapter 2.



A	F
0	1
1	0

Figure 3.5: Symbol for NOT Gate

Table 3.3: Truth Table for NOT Gate

3.1.4 NAND Gate

NAND Gate is a combination of AND Gate and inverter. It works opposite to AND Gate i.e. it produces high output when any of its input is low.

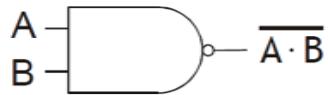


Figure 3.6: Symbol for NAND Gate

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

Table 3.4: Truth Table for NAND Gate

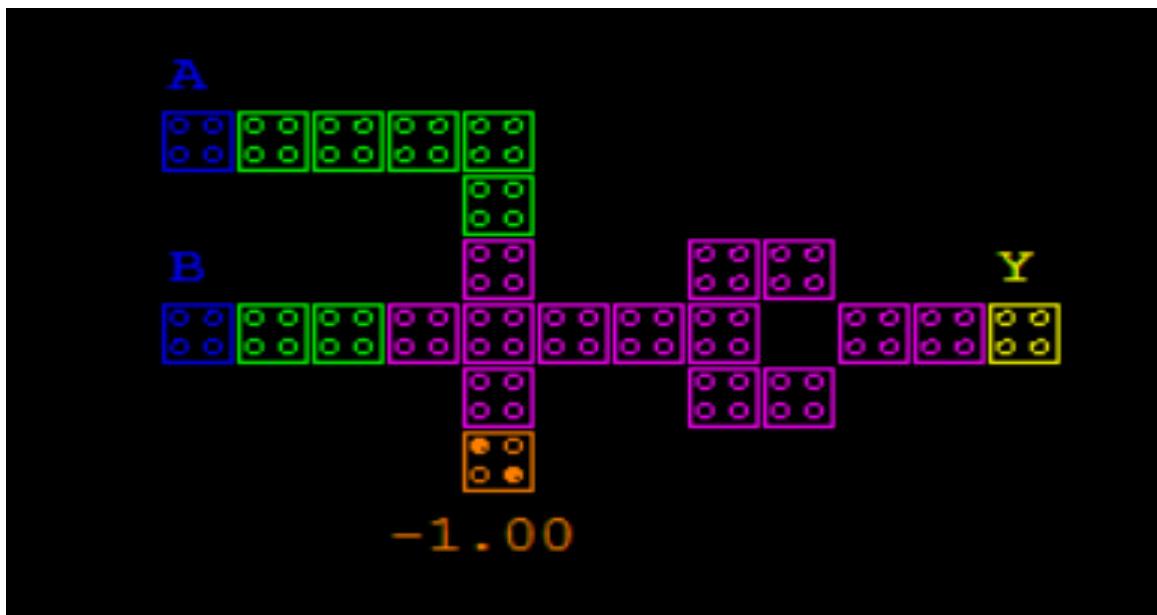


Figure 3.7: QCA Layout for NAND Gate

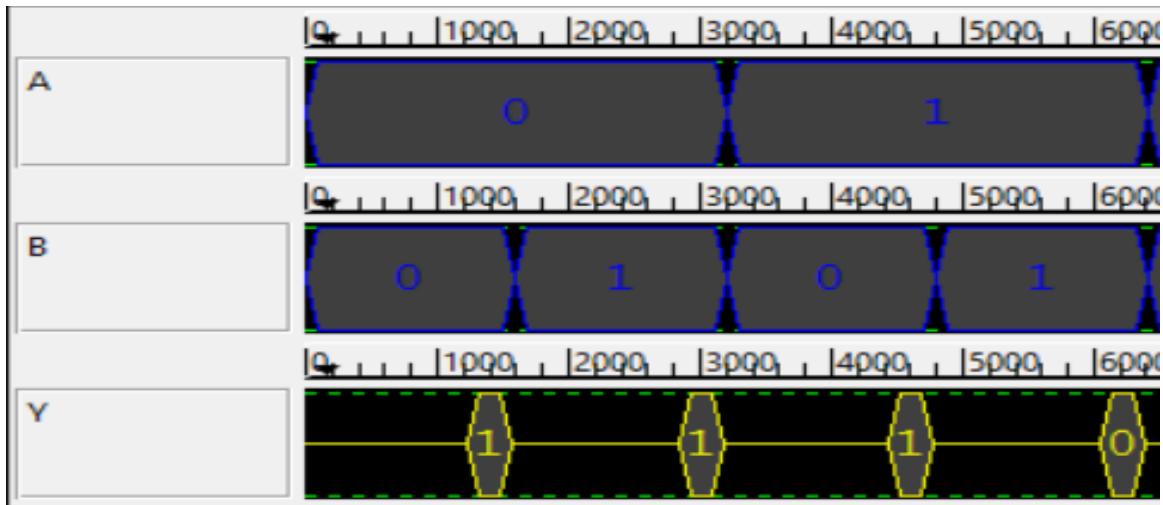


Figure 3.8: Simulation Result for NAND Gate

3.1.5 NOR Gate

NOR Gate is a combination of OR Gate and an Inverter. It produces high output when all of inputs are low.

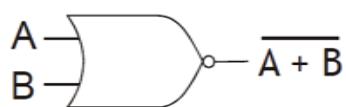


Figure 3.9: Symbol for NOR Gate

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

Table 3.5: Truth Table for NOR Gate

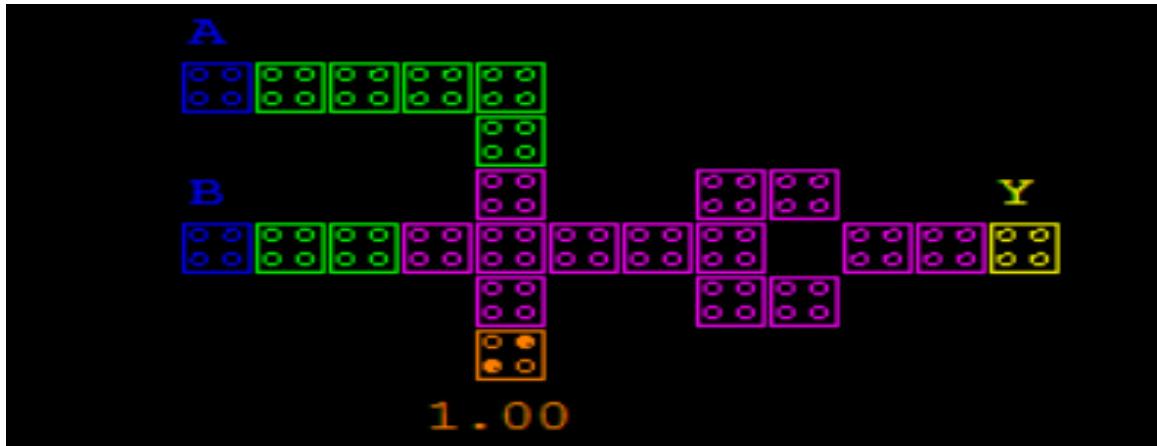


Figure 3.10: QCA Layout for NOR Gate

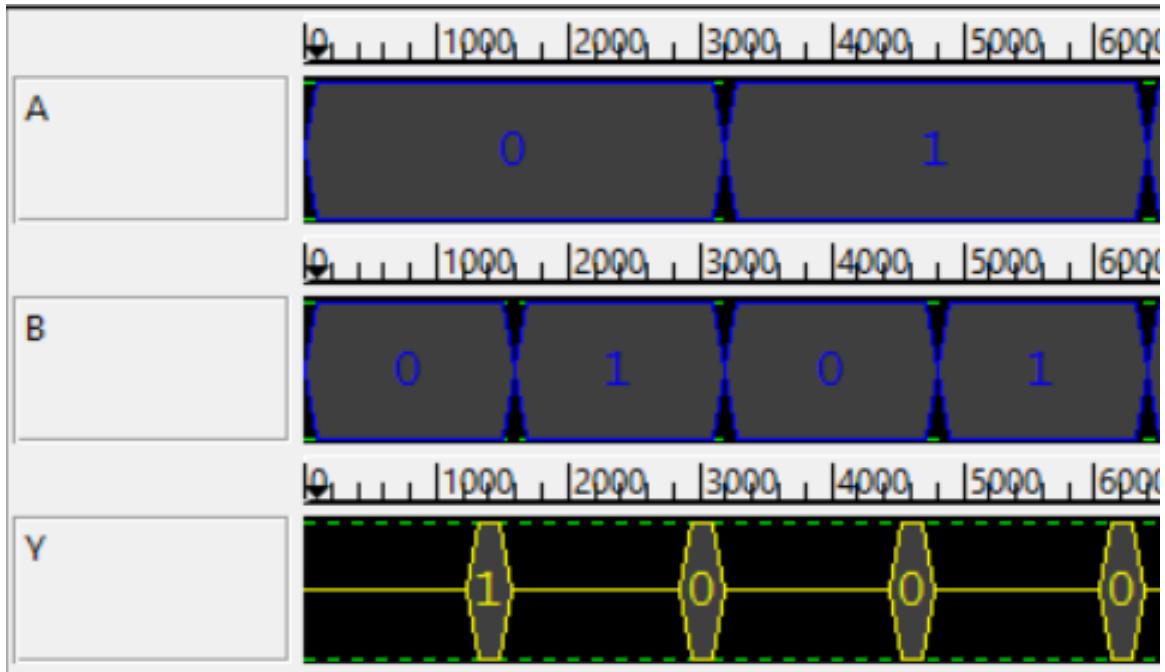


Figure 3.11: Simulation Result for NOR Gate

3.1.6 XOR Gate

This is also known as odd number of 1's detector because it produces high output whenever odd number of inputs are high. Its output is low when all inputs are same. This gate is also known as staircase gate due to the fact that as explained above because we use same logic in stair electrical wiring (If switches at both ends are same, light is *OFF* adn whenever they are at different positions bulb gets switched *ON*). This gate is helpful in logical addition as it does not produce carry when both inputs are "1". Moreover, this is also used in Embedded system to clear ALU due to the fact that whenever a signal is XORED with itself, it results *ZERO* always.

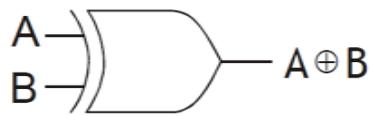
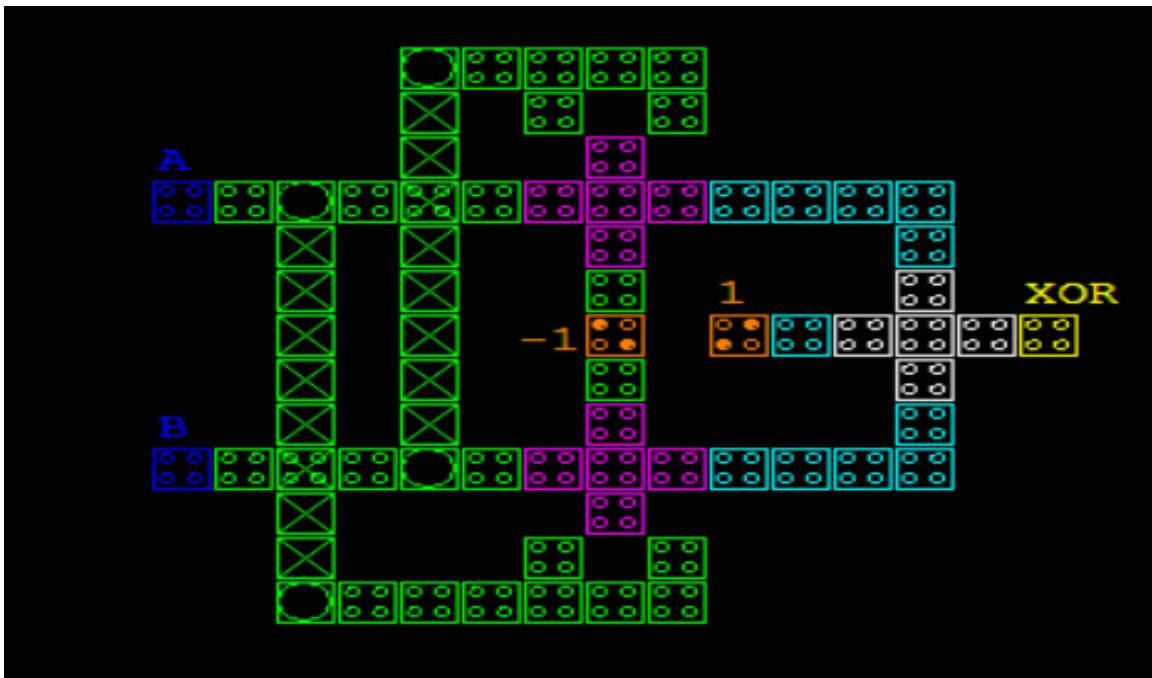


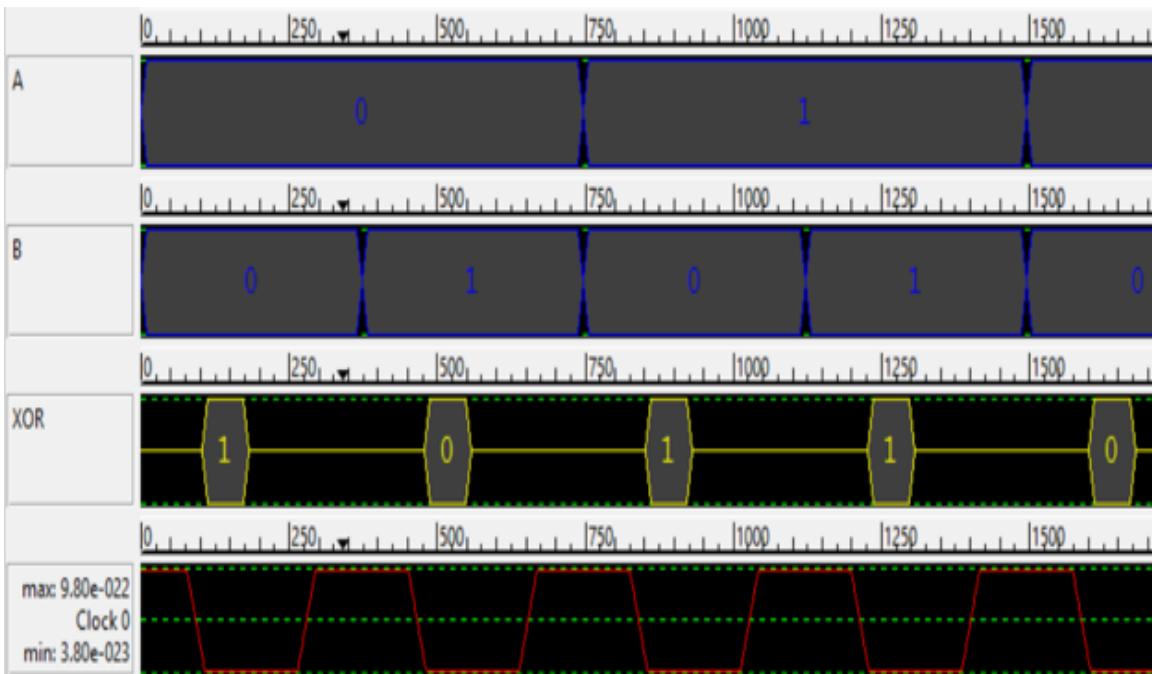
Figure 3.12: Symbol for XOR Gate

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

Table 3.6: Truth Table for XOR Gate



(a) QCA Layout for XOR Gate



(b) Simulation Result for XOR Gate

Figure 3.13: QCA Layout and Simulation Result for XOR Gate

As per Fig. 3.11 we observe that this design gives a Latency (Propagation Delay) of “1” clock pulse. Apart from this design, some other designs for XOR Gate are also implemented and simulated in QCADesigner tool 2.0.3 later in this chapter.

3.1.7 XNOR Gate

It is a XOR Gate followed by an inverter. It is also called as *Coincidence Gate* as it produces high output when all of its inputs are at same level. Also it is called as even number of 1’s detector as its output goes high whenever even number of inputs are high, otherwise it produces low output. Symbol and Truth table for XNOR Gate is shown in Fig. 3.14 and Table 3.7 respectively.

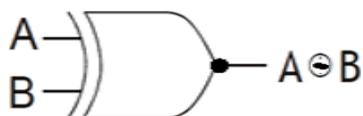


Figure 3.14: Symbol for XNOR Gate

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

Table 3.7: Truth Table for XNOR Gate

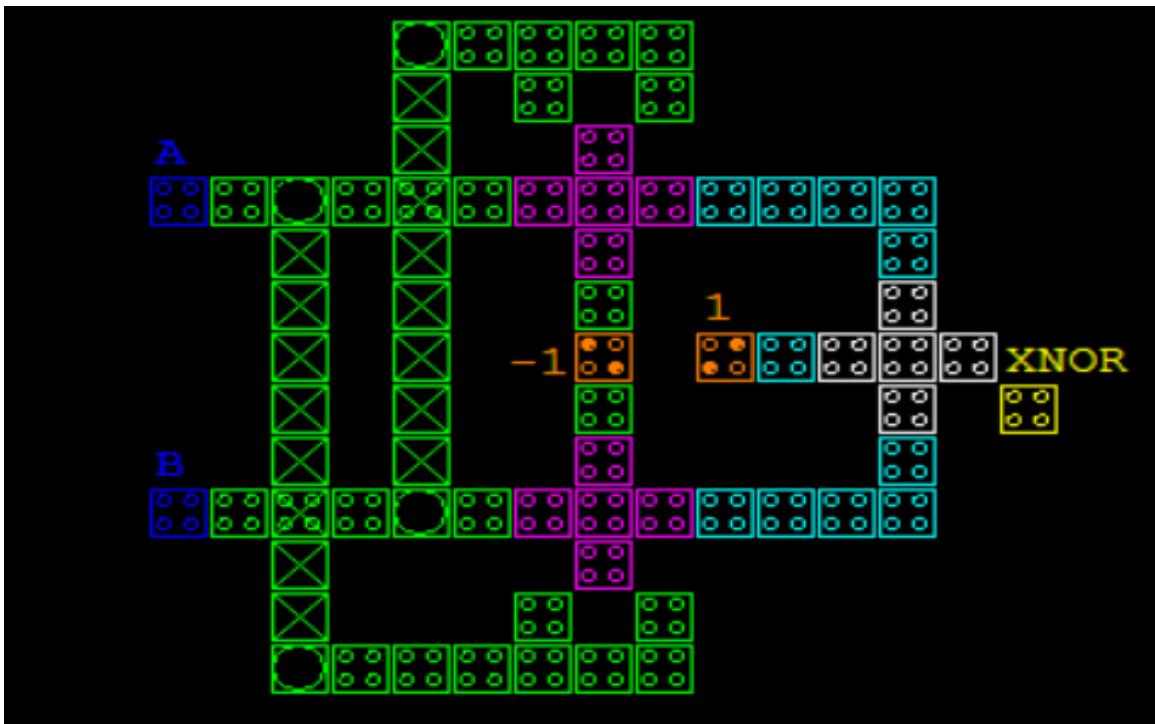


Figure 3.15: QCA Layout for XNOR Gate

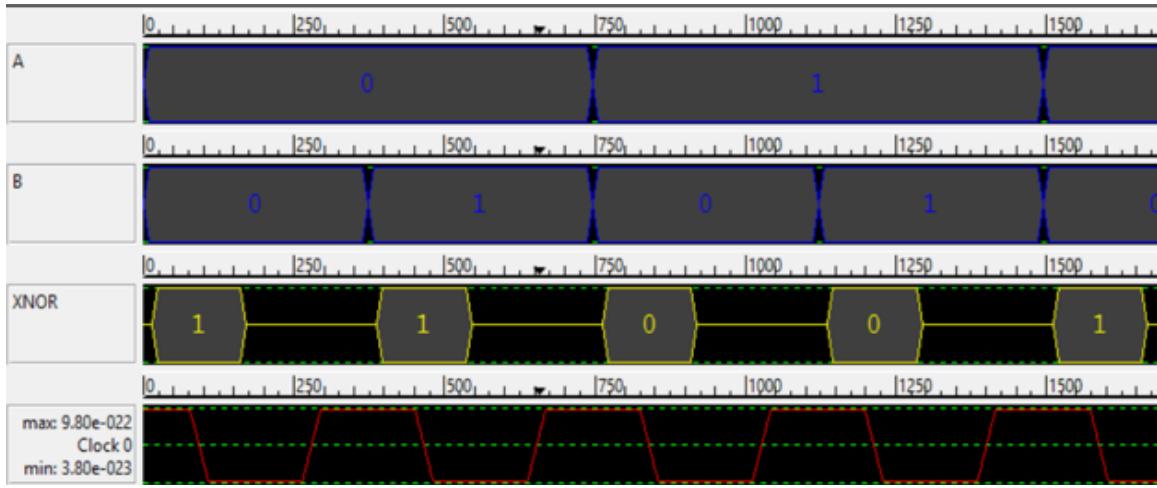


Figure 3.16: Simulation Result for XNOR Gate

Same as XOR Gate, this also has a latency of 1 clock pulse. And this design uses multilayer crossover.

3.2 Different Variants of XOR Gate

Apart from the basic design shown in section 3.1.6, this section consists different versions of XOR gate and their comparison[7][8].

3.2.1 Design-1

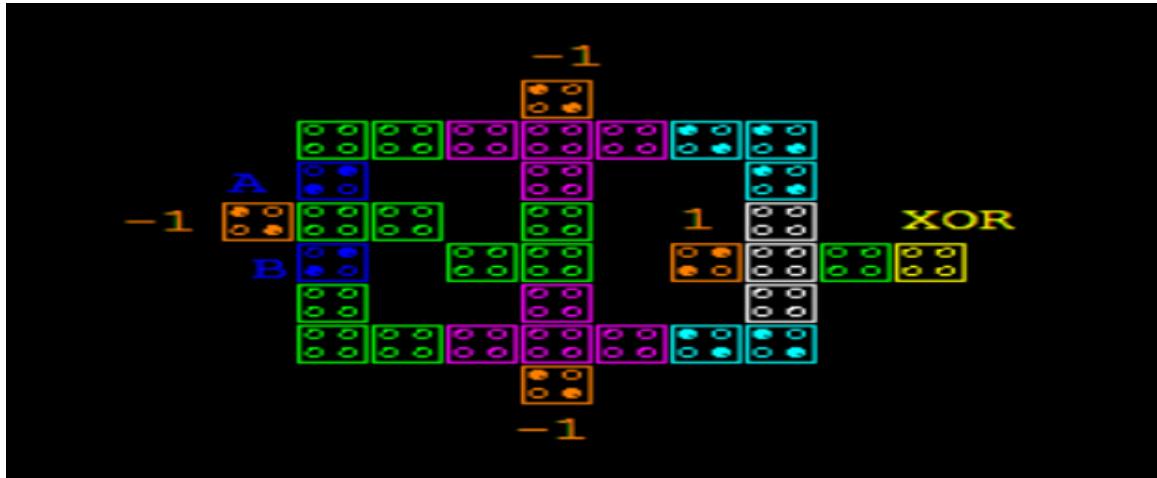


Figure 3.17: QCA Layout for XOR Design 1

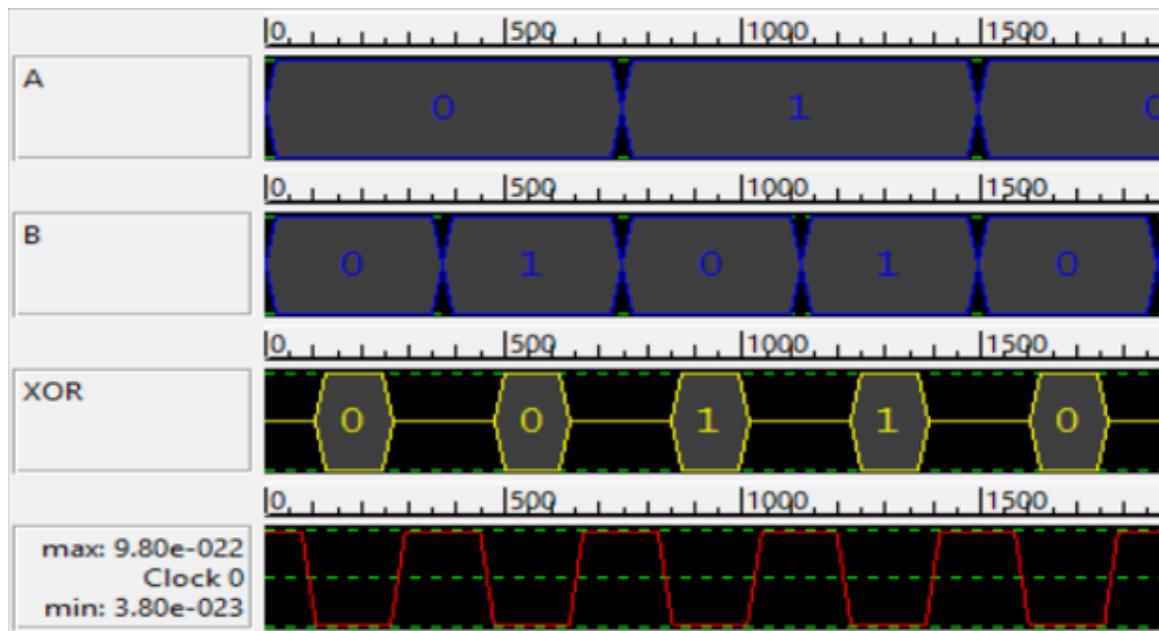


Figure 3.18: Simulation Result XOR Design 1

3.2.2 Design-2

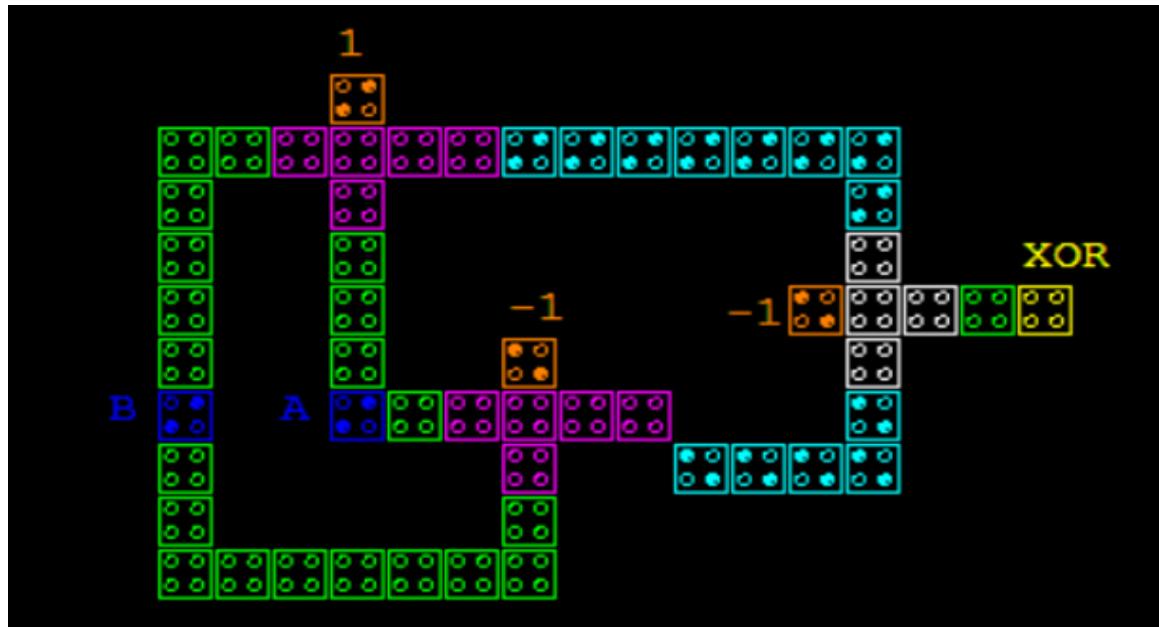


Figure 3.19: QCA Layout for XOR Design 2

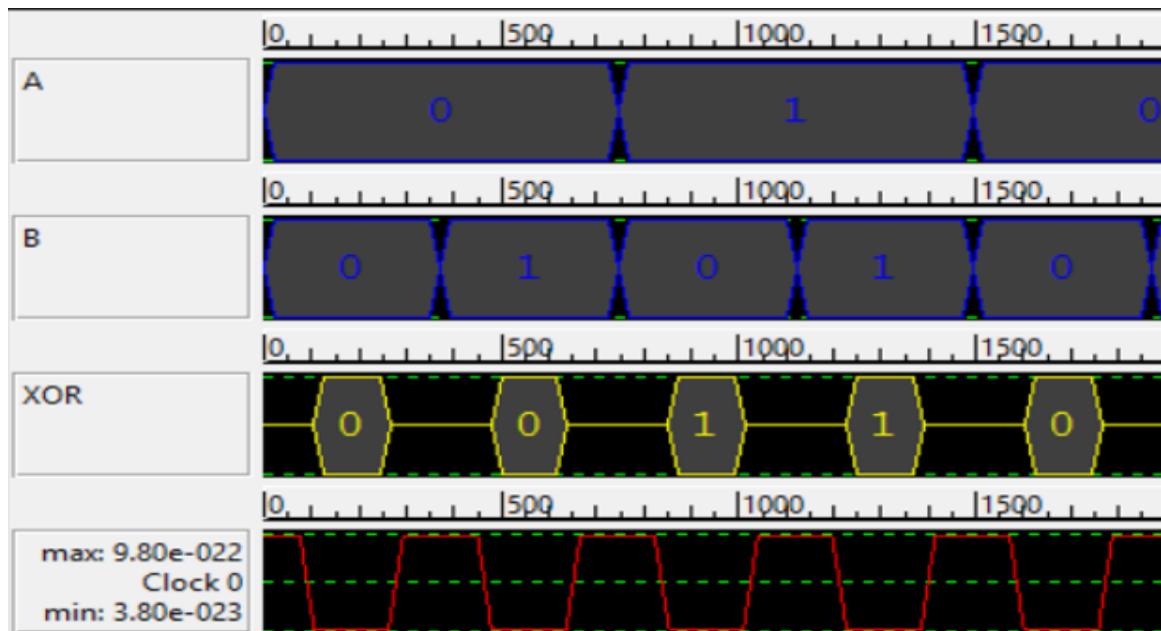


Figure 3.20: Simulation Result XOR Design 2

3.2.3 Design-3

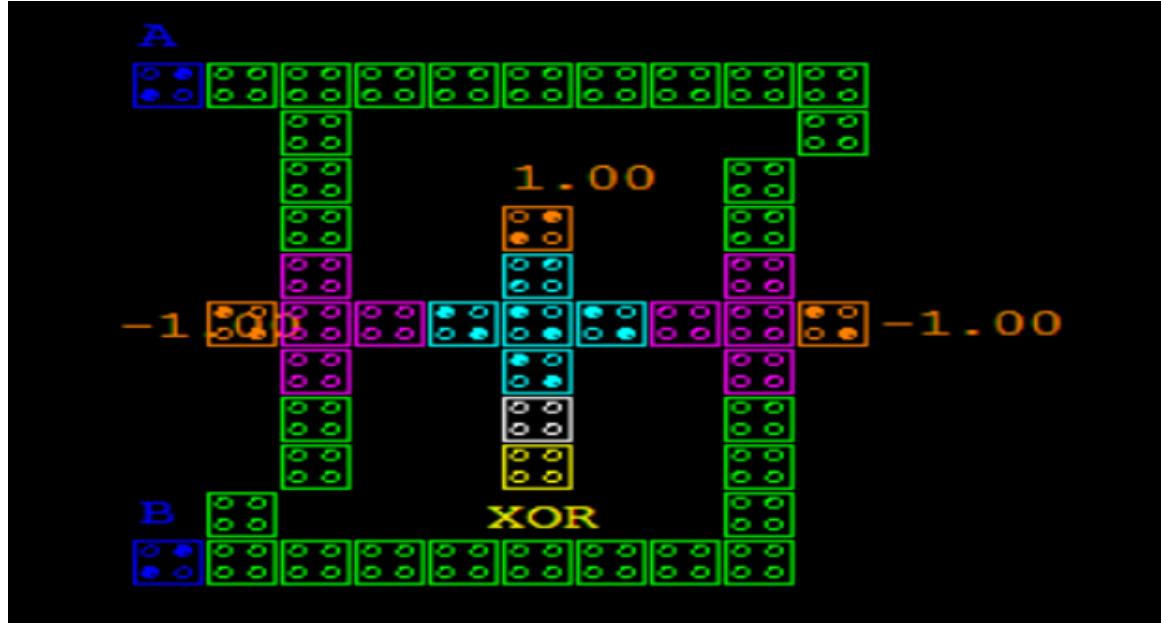


Figure 3.21: QCA Layout for XOR Design 3

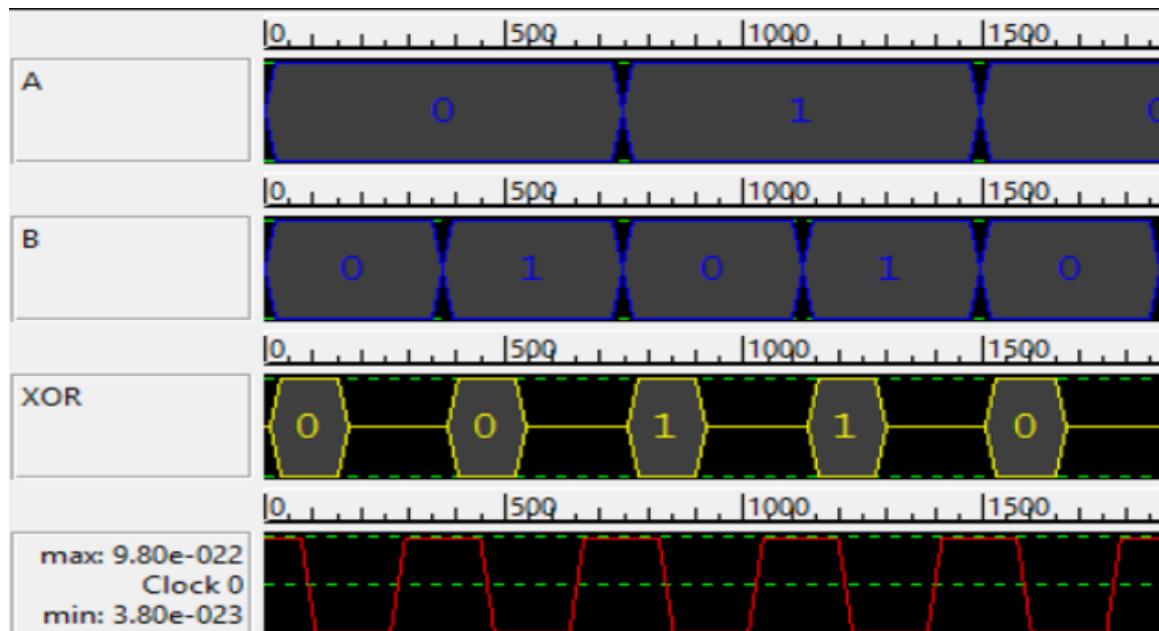


Figure 3.22: Simulation Result XOR Design 3

3.2.4 Design-4

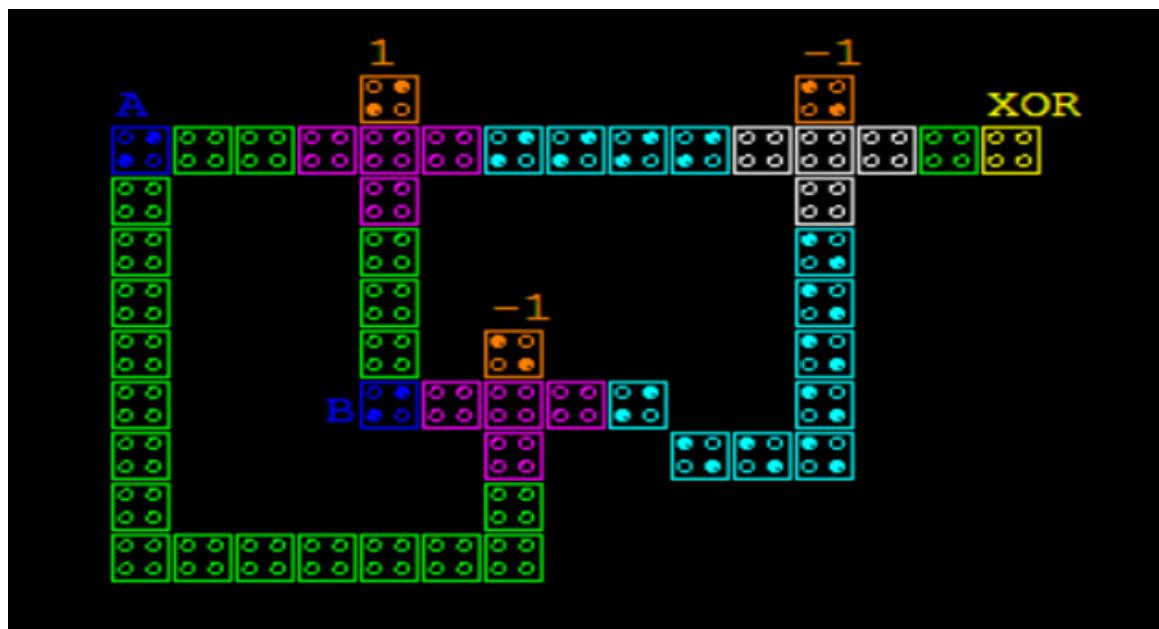


Figure 3.23: QCA Layout for XOR Design 4

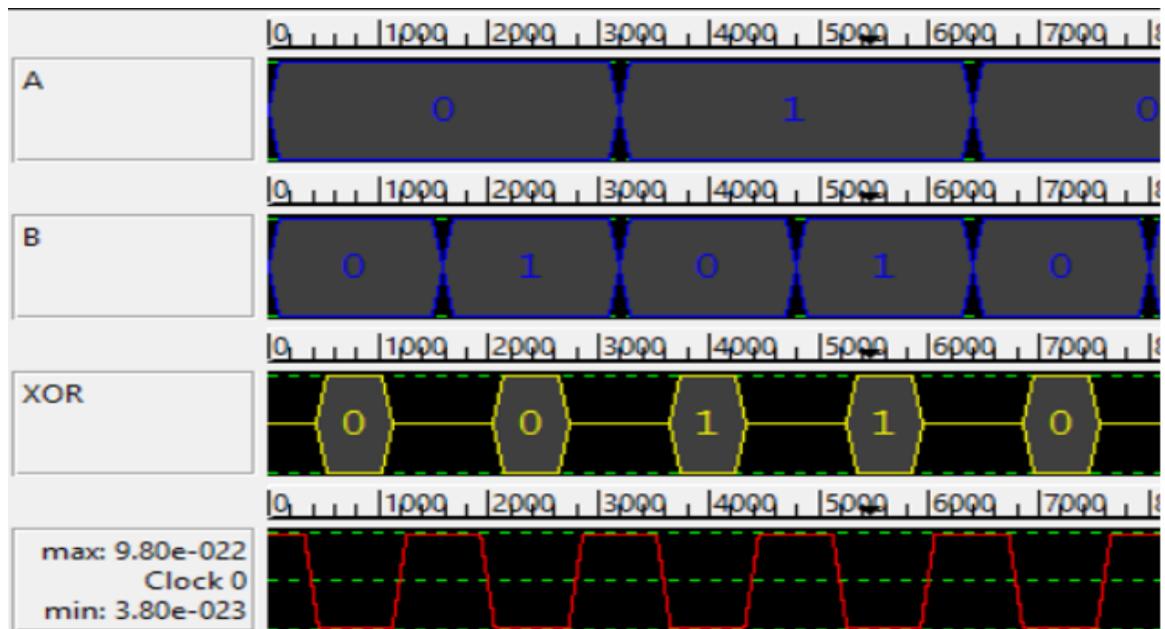


Figure 3.24: Simulation Result XOR Design 4

3.2.5 Design-5

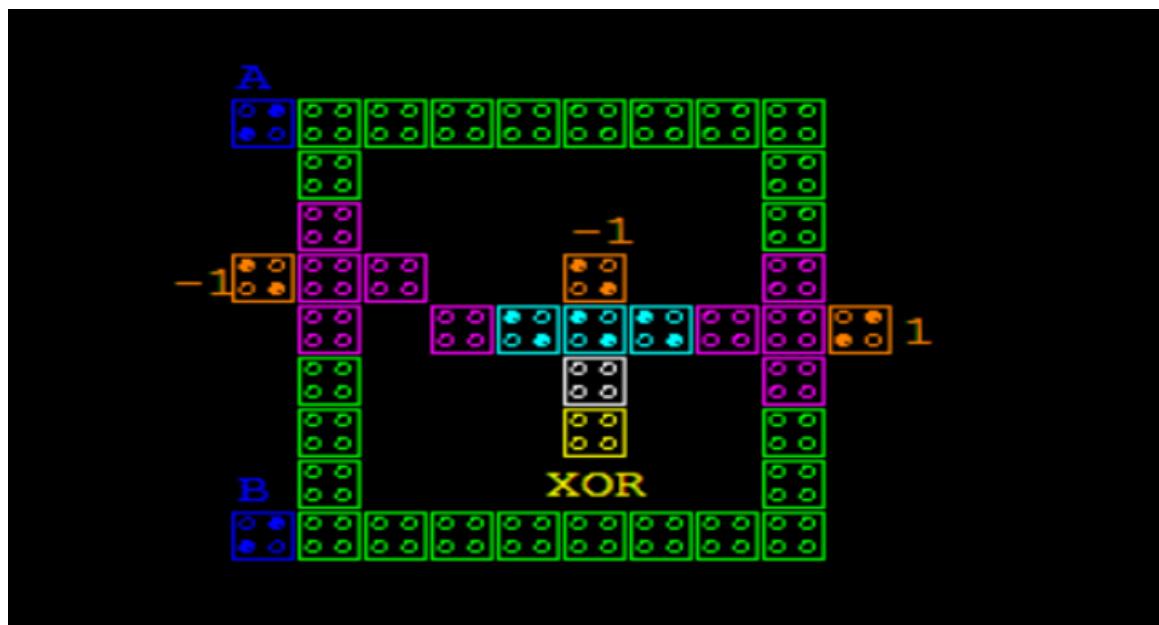


Figure 3.25: QCA Layout for XOR Design 5

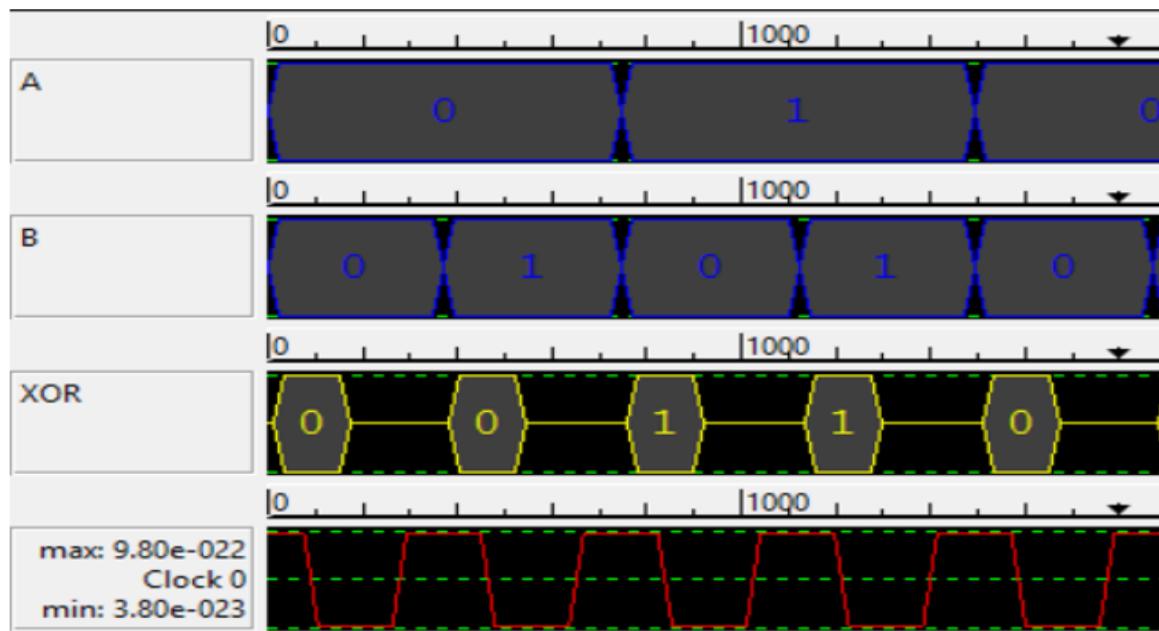


Figure 3.26: Simulation Result XOR Design 5

3.2.6 Design-6

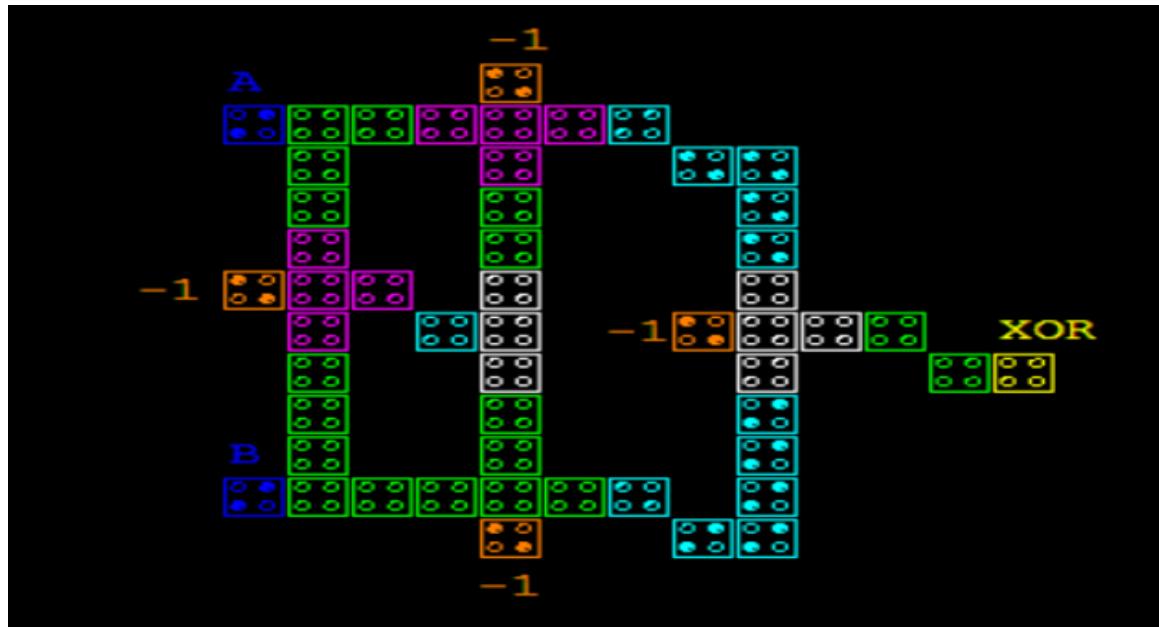


Figure 3.27: QCA Layout for XOR Design 6

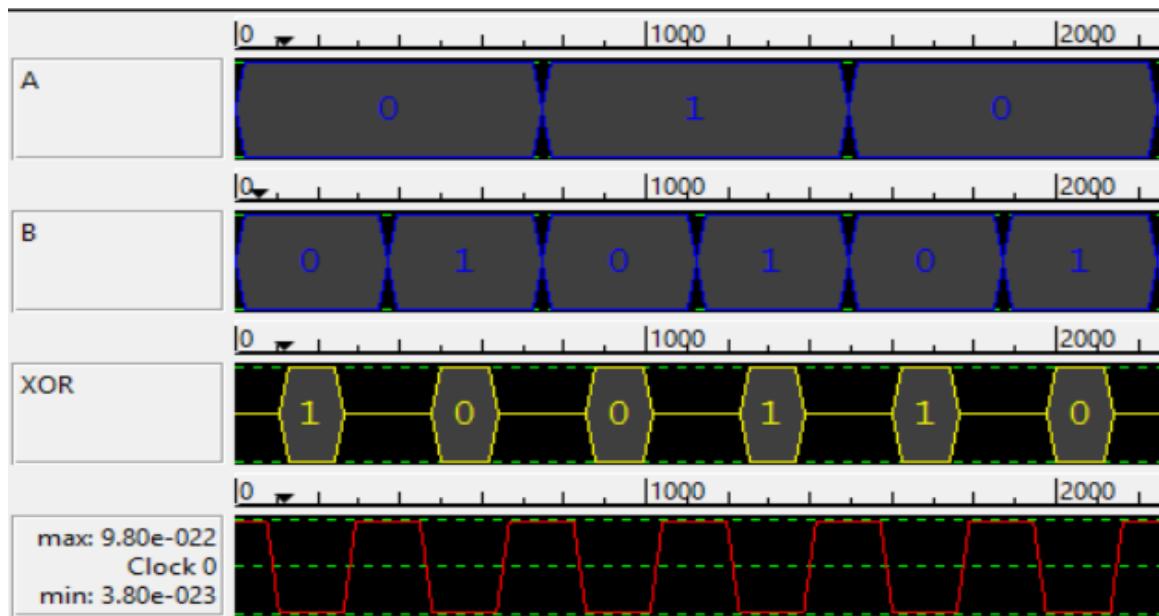


Figure 3.28: Simulation Result XOR Design 6

3.2.7 Design-7

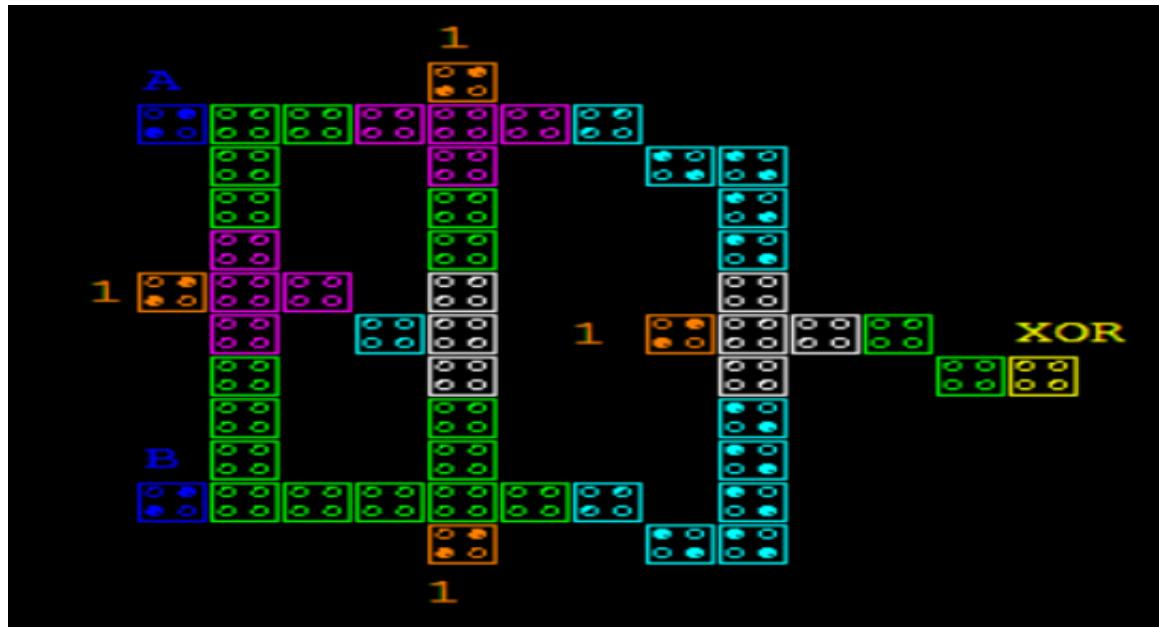


Figure 3.29: QCA Layout for XOR Design 7

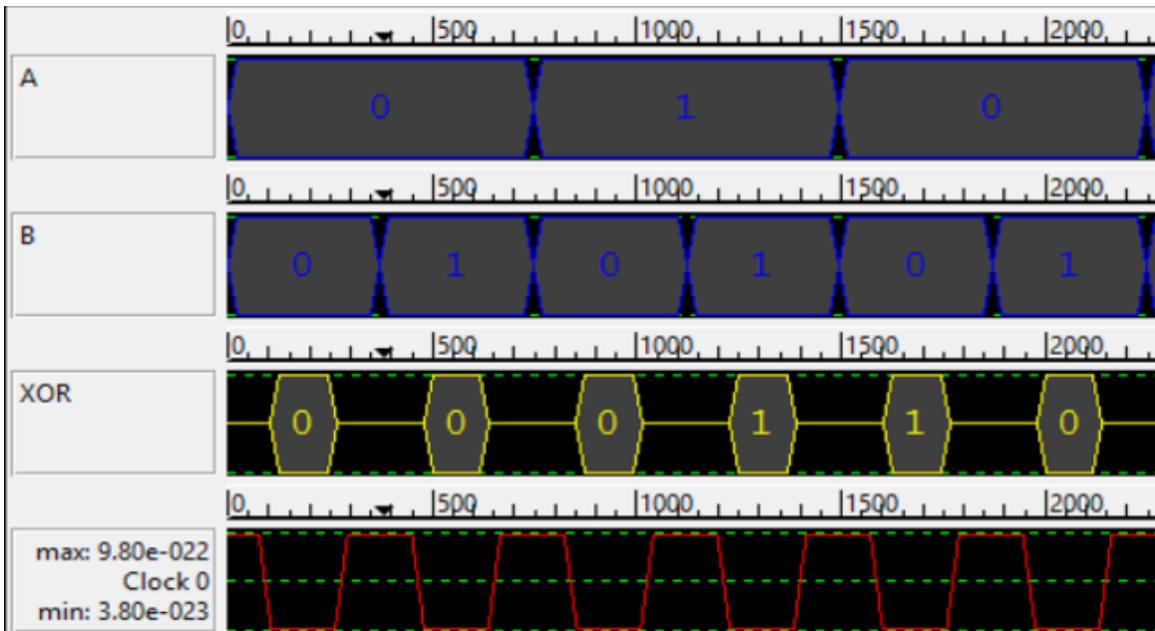


Figure 3.30: Simulation Result XOR Design 7

3.2.8 Performance Analysis of XOR Designs

	Number of Cells	Area	Latency (Clock Cycle)	Simulation Time (Sec)	Crossover (Y/N)	Crossover Type
Basic	83	0.08	1	48	Y	Multi-Layer
Design-1	34	0.05	1	17	N	N/A
Design-2	54	0.08	1	27	N	N/A
Design-3	49	0.06	0.5	25	N	N/A
Design-4	51	0.07	1	25	N	N/A
Design-5	43	0.05	0.5	21	N	N/A
Design-6	52	0.09	2	26	N	N/A
Design-7	52	0.08	2	30	N	N/A

Table 3.8: XOR Designs Comparative Study

All these designs are designed and simulated in QCADesigner tool 2.0.3 with Coherence Vector Simulation Engine. The parameters of simulation are as follow

Temperature:	1.000000	K
Relaxation Time:	1.000000e-015	s
Time Step:	1.000000e-016	s
Total Simulation Time:	7.000000e-011	s
Clock High:	9.800000e-022	J
Clock Low:	3.800000e-023	J
Clock Shift:	0.000000e+000	
Clock Amplitude Factor:	2.000000	
Radius of Effect:	80.000000	nm
Relative Permittivity:	12.900000	
Layer Separation:	11.500000	nm

Figure 3.31: Coherence Vector Simulation Engine Parameters

3.3 MUX Designing

Multiplexer (MUX) is a universal combinational circuit. It is named so because we can implement any of the logic gate by using this. It generally has inputs in 2^n form with one output and 'n' select lines. Select lines are used to determine which signal will be transmitted on output line. Logic for 2×1 MUX is shown as below[8].

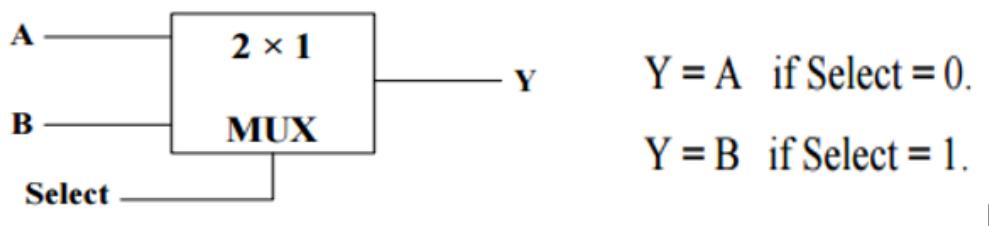


Figure 3.32: Basic Symbol for 2×1 MUX

As shown in table 3.9, it is clear that input A is transmitted when select line is at low level and B is transmitted when Select line is high. By using these results, we can define MUX function as[24]

$$f(A, B, \text{Select}) = AS' + BS$$

Select	A	B	Y
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

Table 3.9: Truth Table for 2×1 MUX

3.3.1 MUX Design-1

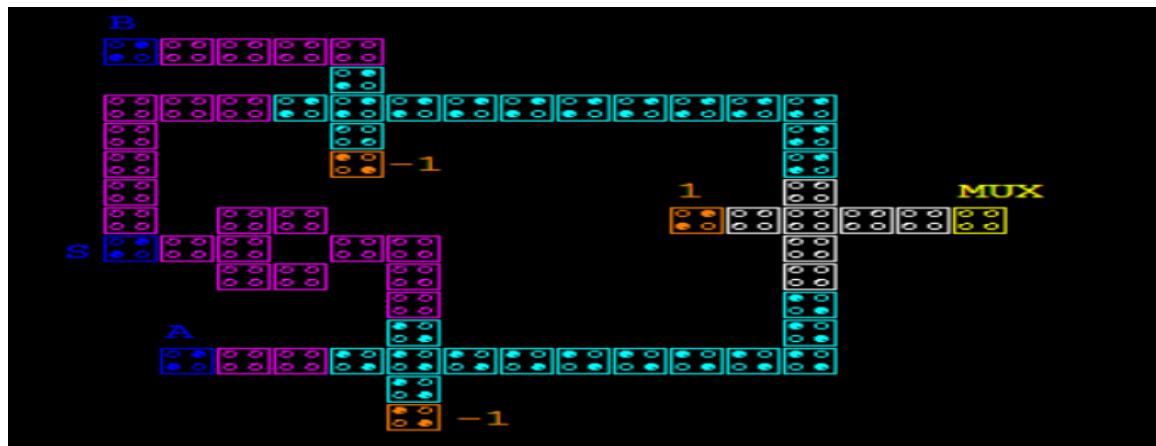


Figure 3.33: QCA Layout for MUX Design 1

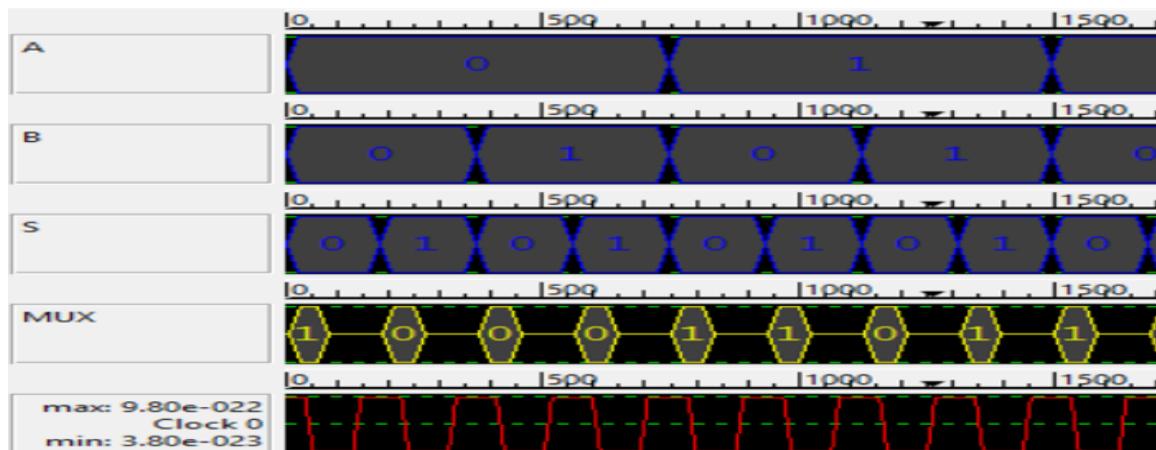


Figure 3.34: Simulation Result MUX Design 1

3.3.2 MUX Design-2

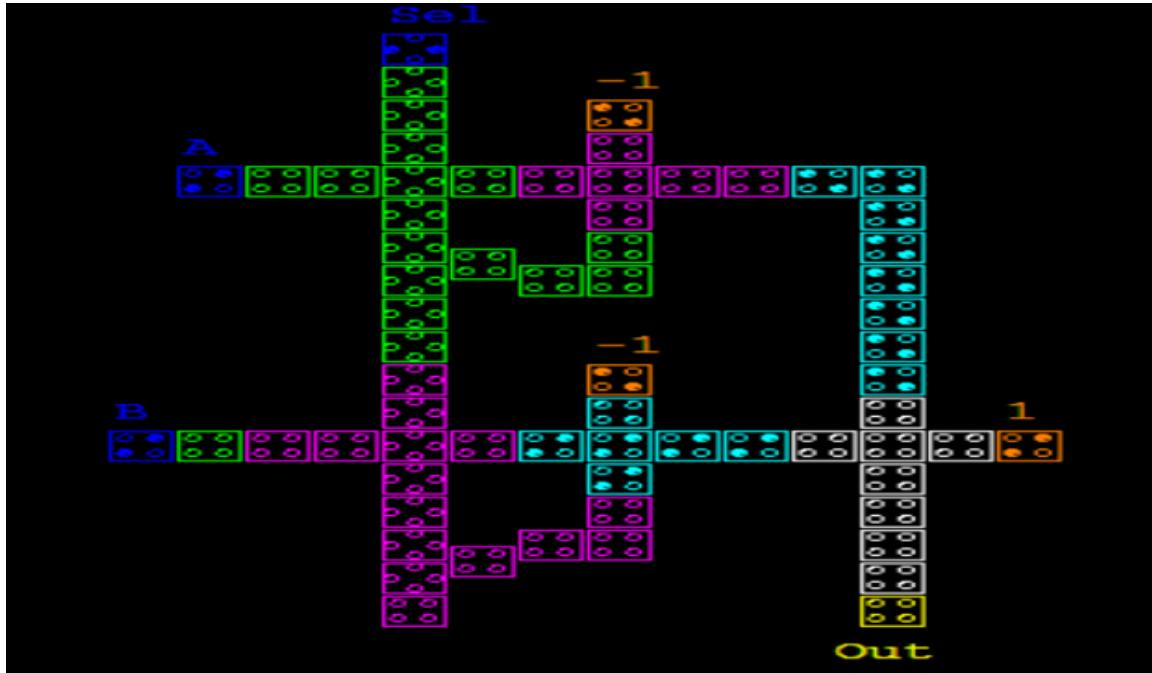


Figure 3.35: QCA Layout for MUX Design 2

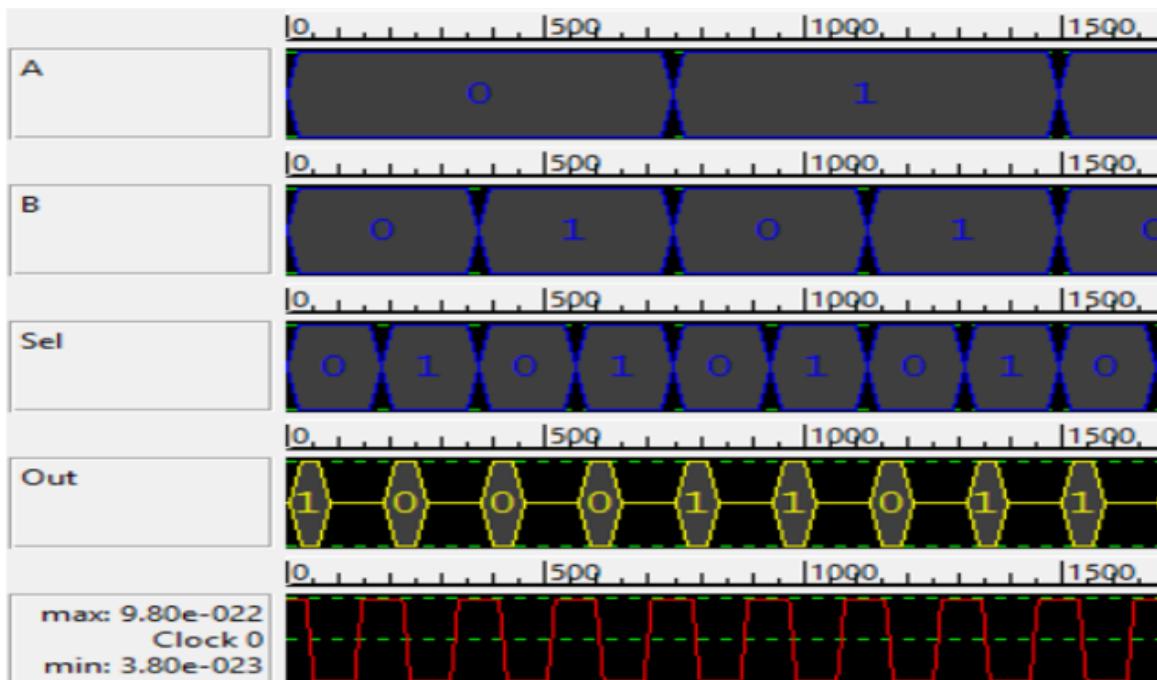


Figure 3.36: Simulation Result MUX Design 2

3.3.3 MUX Design-3

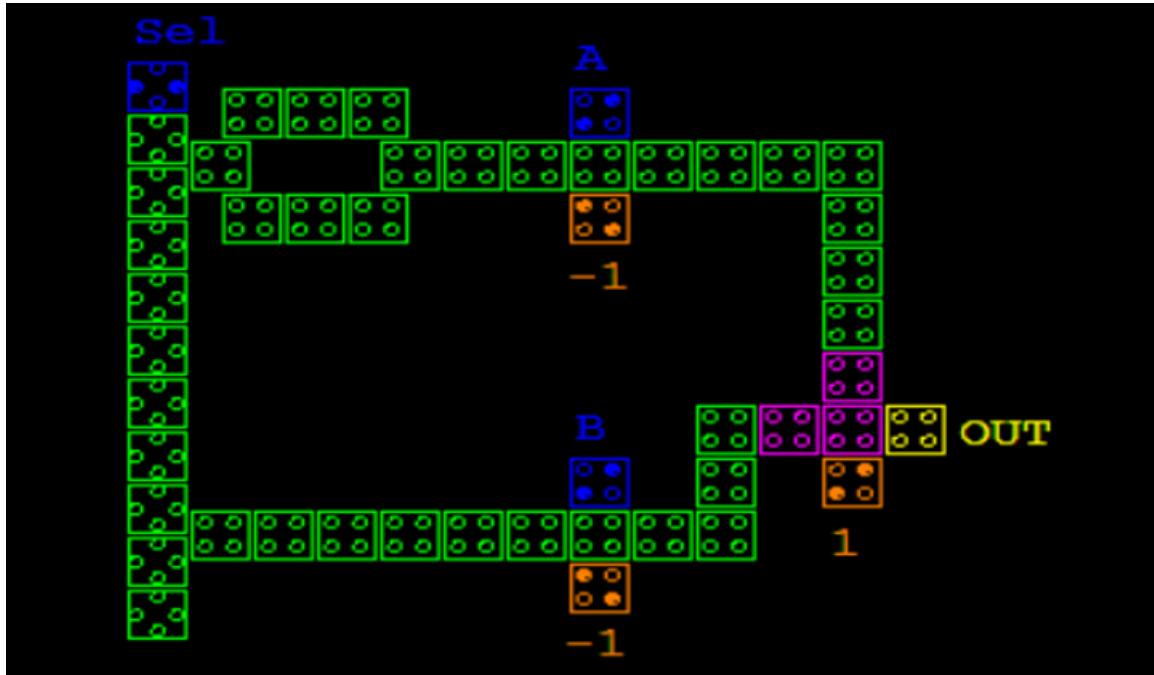


Figure 3.37: QCA Layout for MUX Design 3

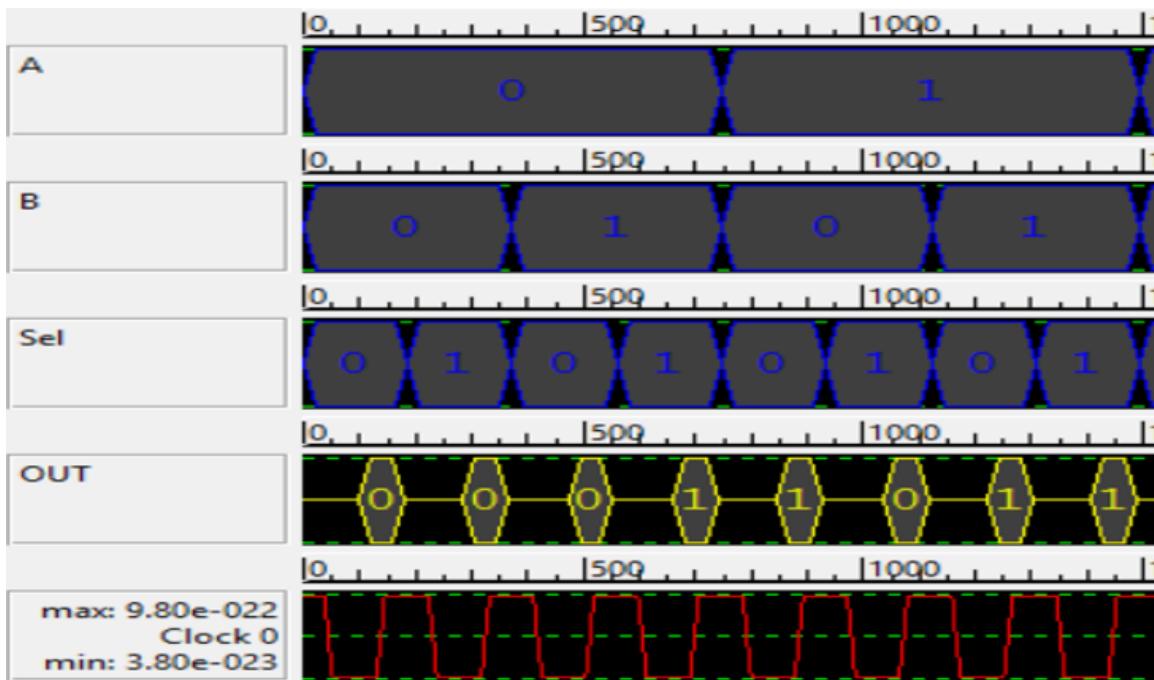


Figure 3.38: Simulation Result MUX Design 3

3.3.4 MUX Design-4

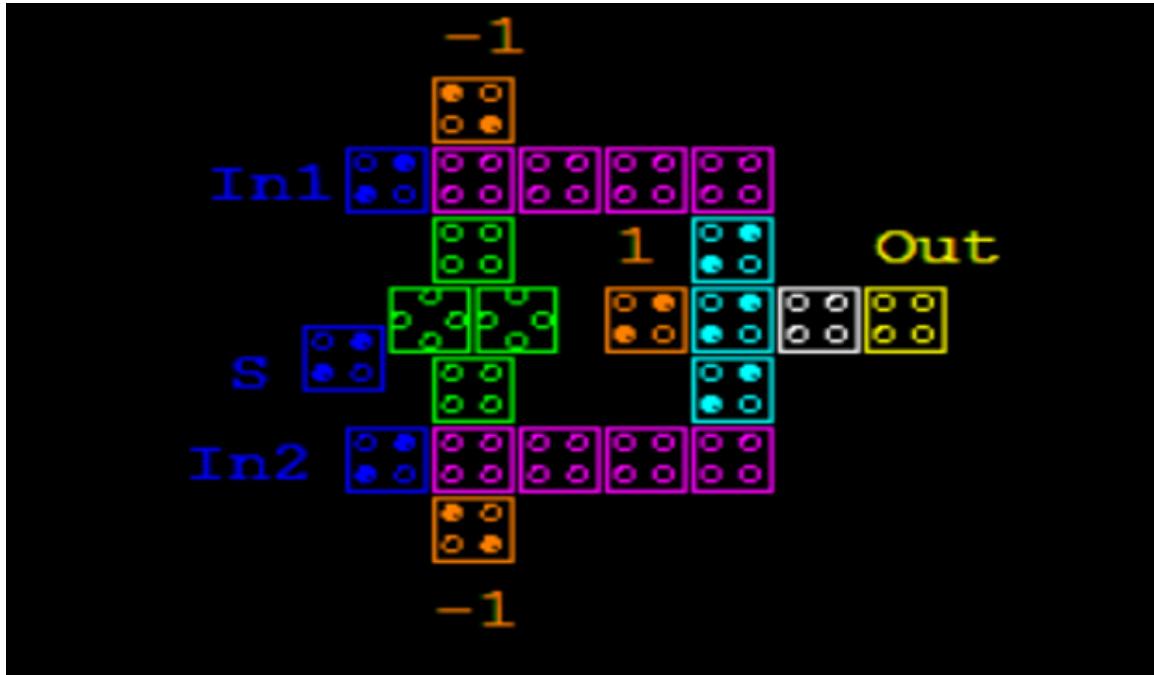


Figure 3.39: QCA Layout for MUX Design 4

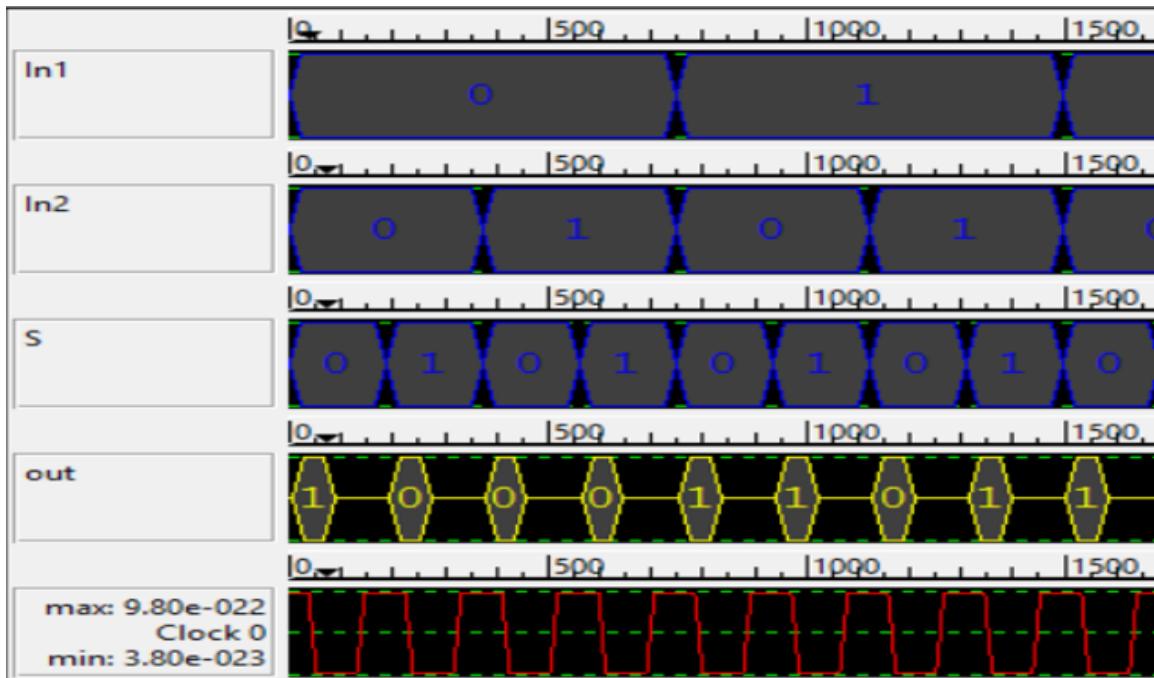


Figure 3.40: Simulation Result MUX Design 4

3.3.5 MUX Design-5

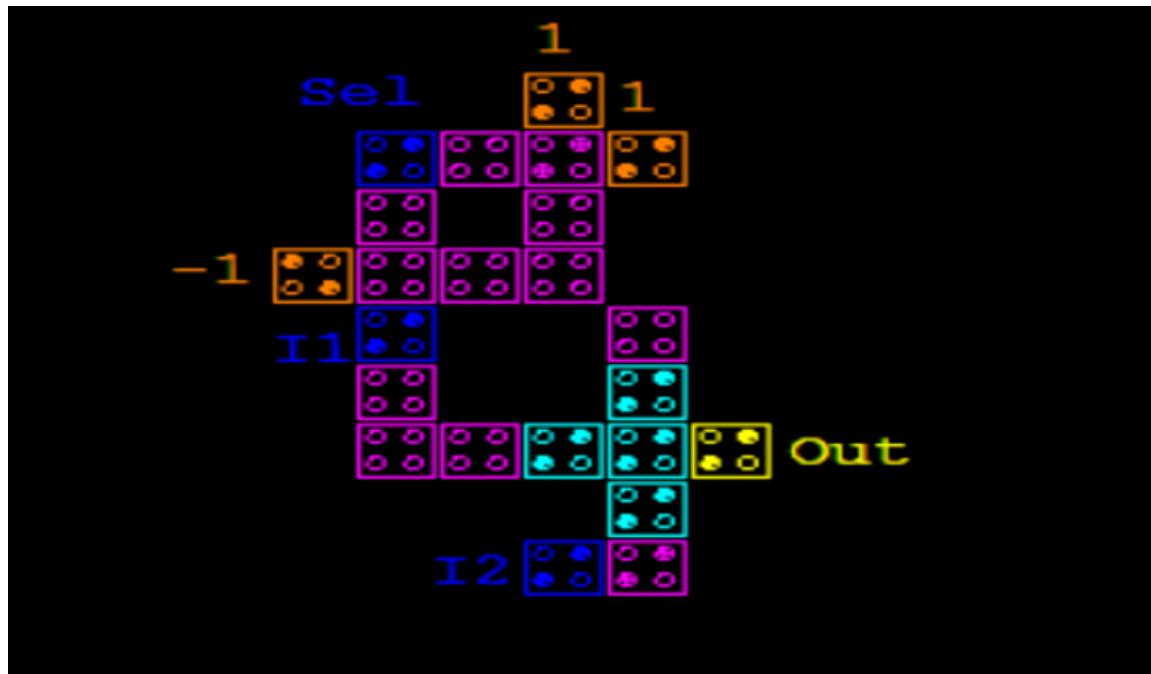


Figure 3.41: QCA Layout for MUX Design 5

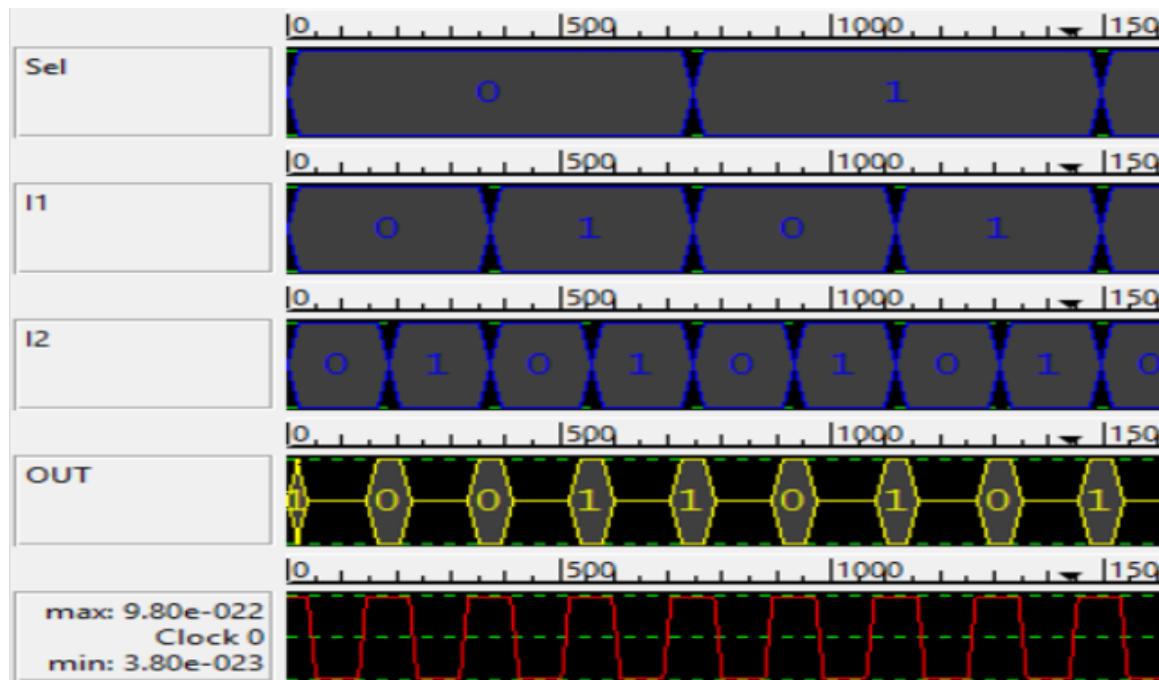


Figure 3.42: Simulation Result MUX Design 5

3.3.6 Comparison of 2×1 MUX Designs

	Number of Cells	Area	Latency (Clock Cycle)	Simulation Time (Sec)
MUX Design-1	64	0.11	1	32
MUX Design-2	67	0.11	1	37
MUX Design-3	49	0.07	0	25
MUX Design-4	23	0.04	1	10
MUX Design-5	23	0.04	0.75	10

Table 3.10: Comparative Study of 2×1 MUX Desings

3.3.7 Designing of 4×1 MUX by using 2×1 MUX

Higher order MUX can be implemented by using lower order MUX. For example, we can use three 2×1 MUX to implement a 4×1 MUX by using the following logic:

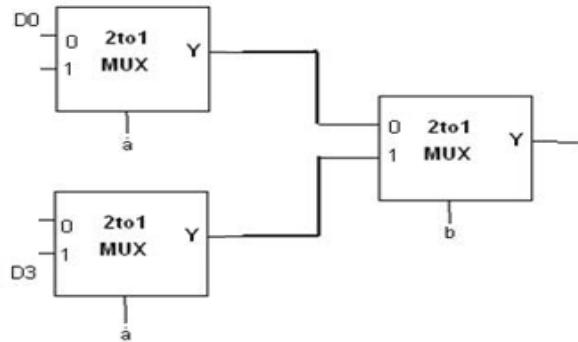
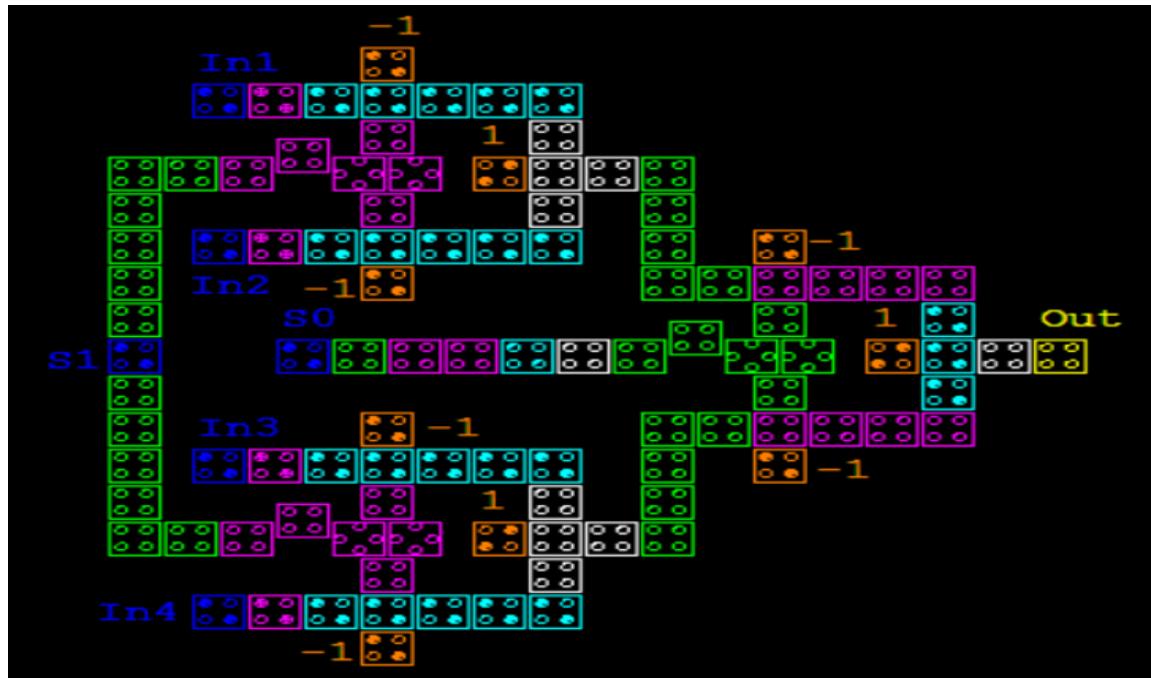


Figure 3.43: Basic Symbol 4×1 MUX using 2×1 MUX

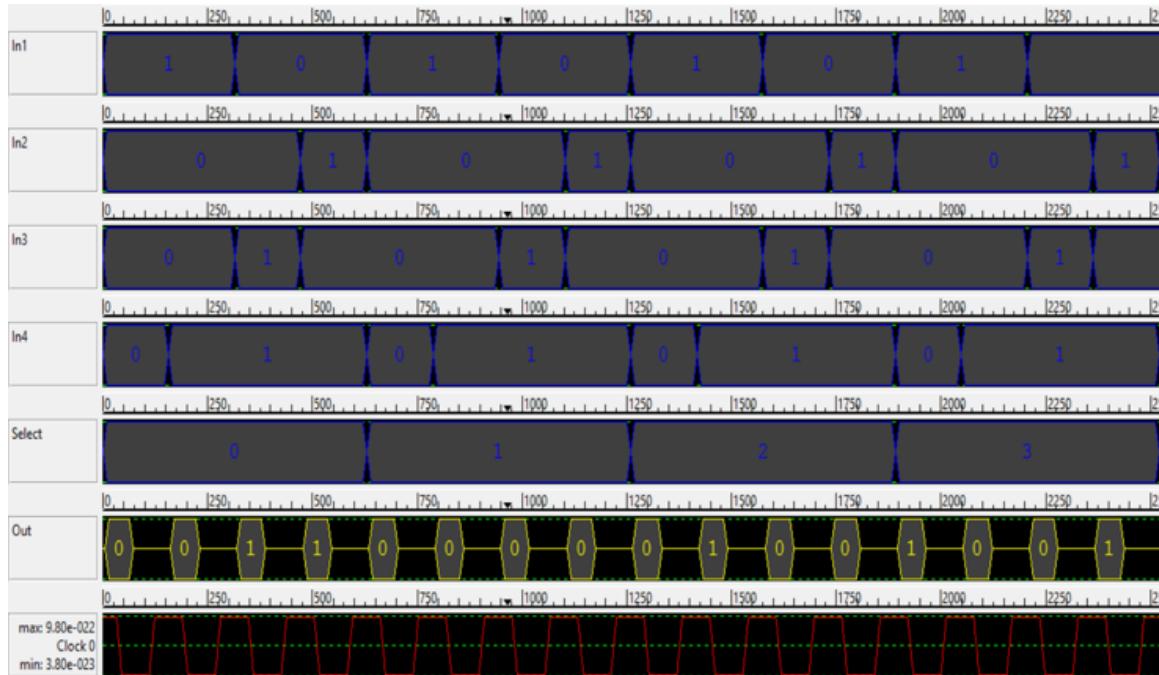
As shown in Fig. 3.43, a and b are select lines and $D0 - D3$ are data inputs. Since here are 4 inputs and 1 output hence it's a 4×1 MUX. We may also calculate required 2×1 MUX to design 4×1 as:

$$N = \frac{4}{2} = \frac{2}{2}$$

by adding quotients of both stages $2 + 1 = 3$. QCA Layout and simulation result for 4×1 MUX using 2×1 MUX design is shown below:



(a) QCA Layout



(b) Simulation Result

Figure 3.44: QCA Layout and Simulation Result for 4*1 MUX using 2*1 MUX

3.4 Adder Designing

Adders are combinational circuits used to compute addition of inputs along with carry. There are different types of adders present. Few of them are implemented & verified using QCADesigner tool 2.0.3.

3.4.1 1-Bit Adders

1 bit adder is used to compute addition of three bits out of which two are input bits (Say A and B here) and third is previous carry (Carry generated from previous add operation, known as C_{in}). Firstly, two inputs A & B are applied to data inputs and third input is taken from carry flag. According to previous operation result, carry flag is either Set (Logic “1”) or Reset (Logic “0”) and so does C_{in} . After the computation, depending upon the result, flag register will either be Set or Reset[15][16][17][18].

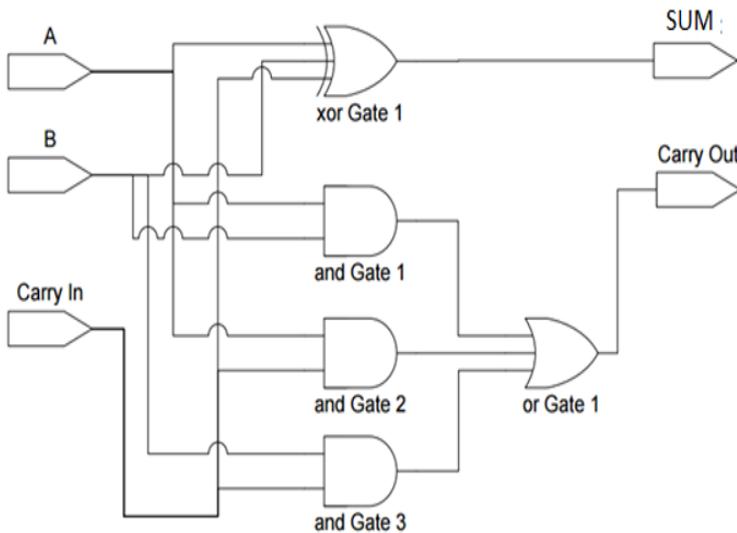


Figure 3.45: Circuit Diagram for 1-bit Adder

By the Fig. 3.45 we can define SUM and Carry Out functions as below:

$$SUM = A \oplus B \oplus C$$

$$C_{Out} = AB + BC + AC$$

Based upon these results, We have tried to implement and simulate some of the existing adder designs. Simulation is done using QCADesigner tool 2.0.3 with all parameters as of Figure 3.31.

3.4.1.1 Design-1

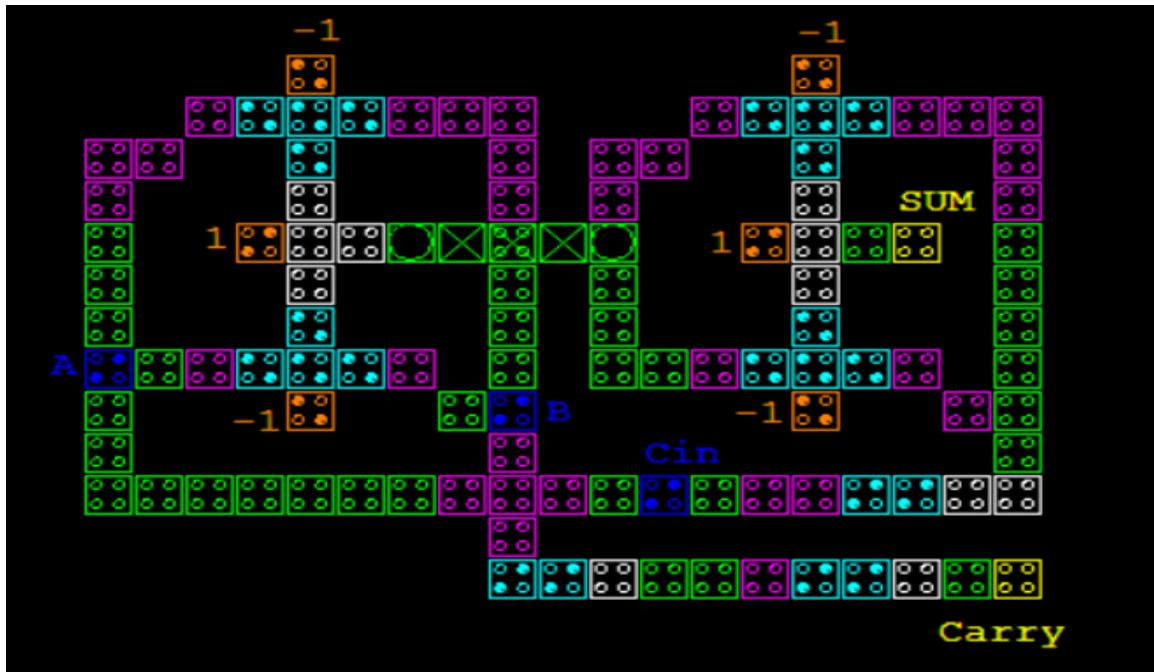


Figure 3.46: QCA Layout for Adder Design 1

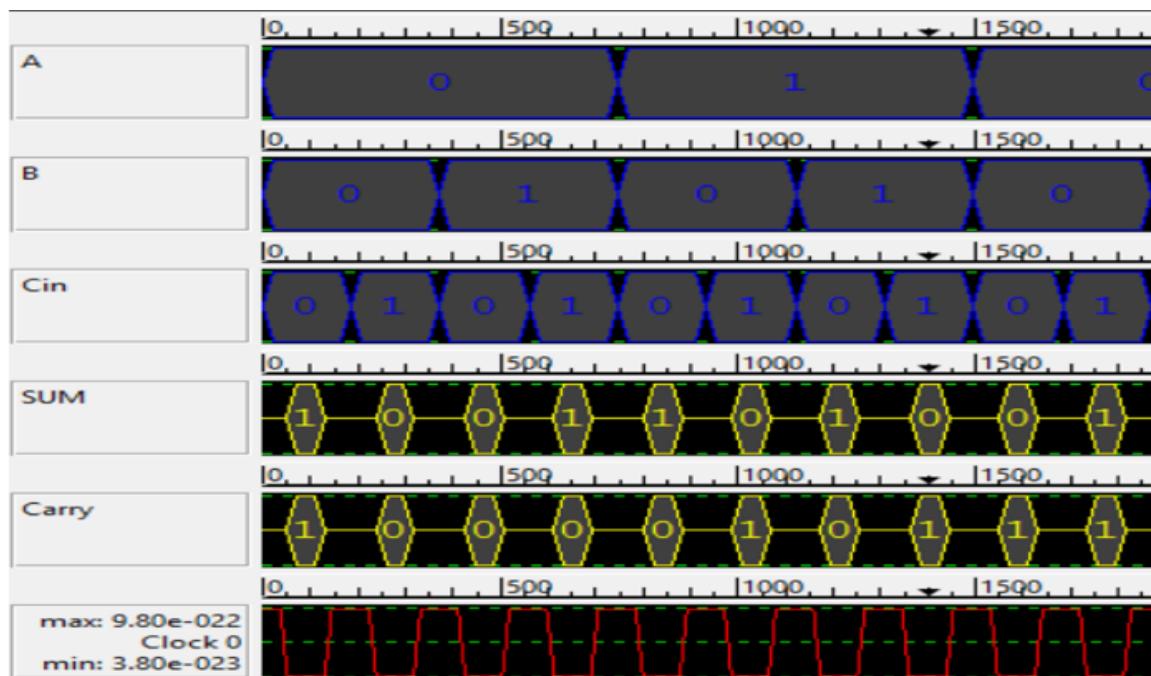


Figure 3.47: Simulation Result Adder Design 1

3.4.1.2 Design-2

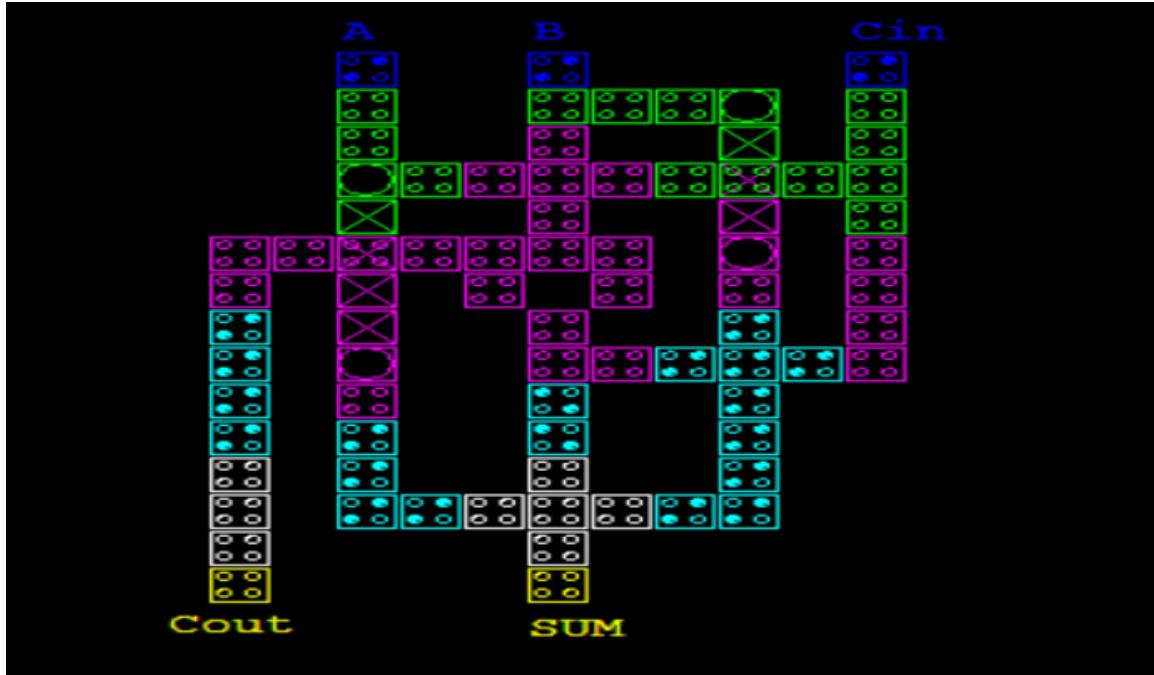


Figure 3.48: QCA Layout for Adder Design 2

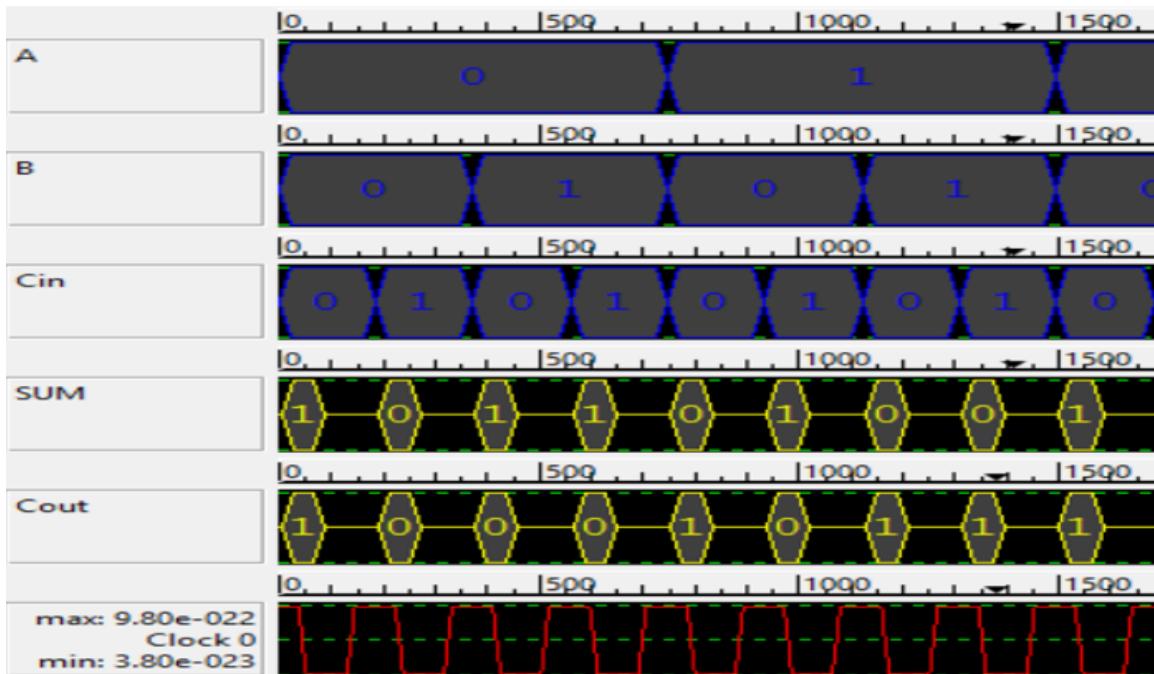


Figure 3.49: Simulation Result Adder Design 2

3.4.1.3 Design-3 (Multi-Layer Design)

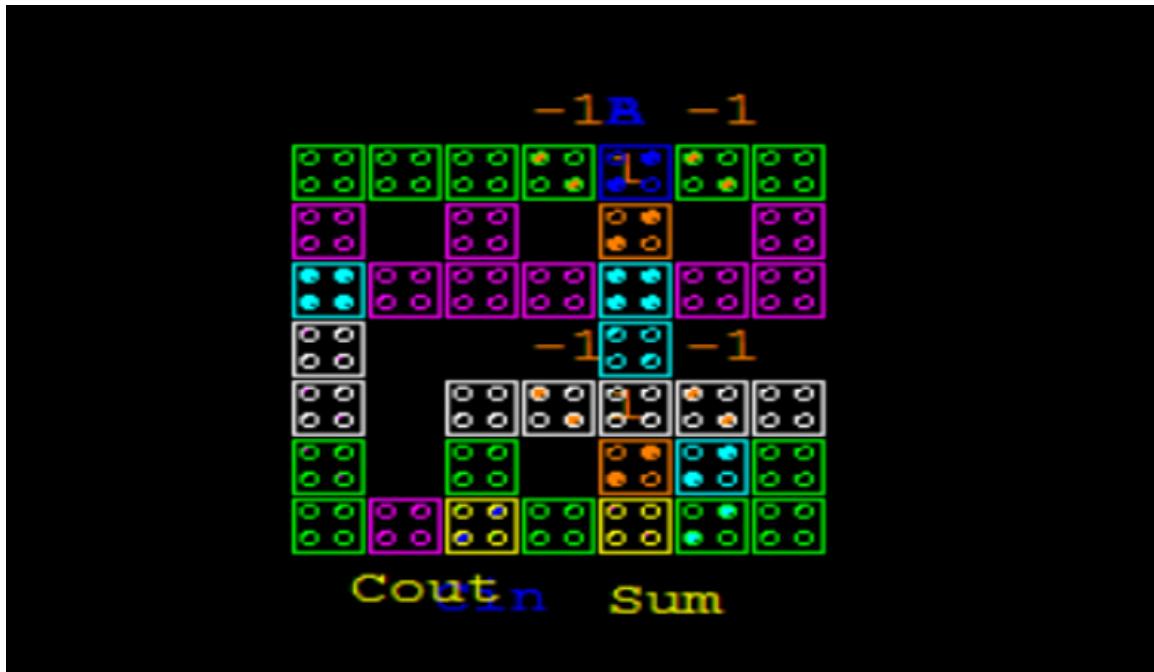


Figure 3.50: QCA Layout for Adder Design 3

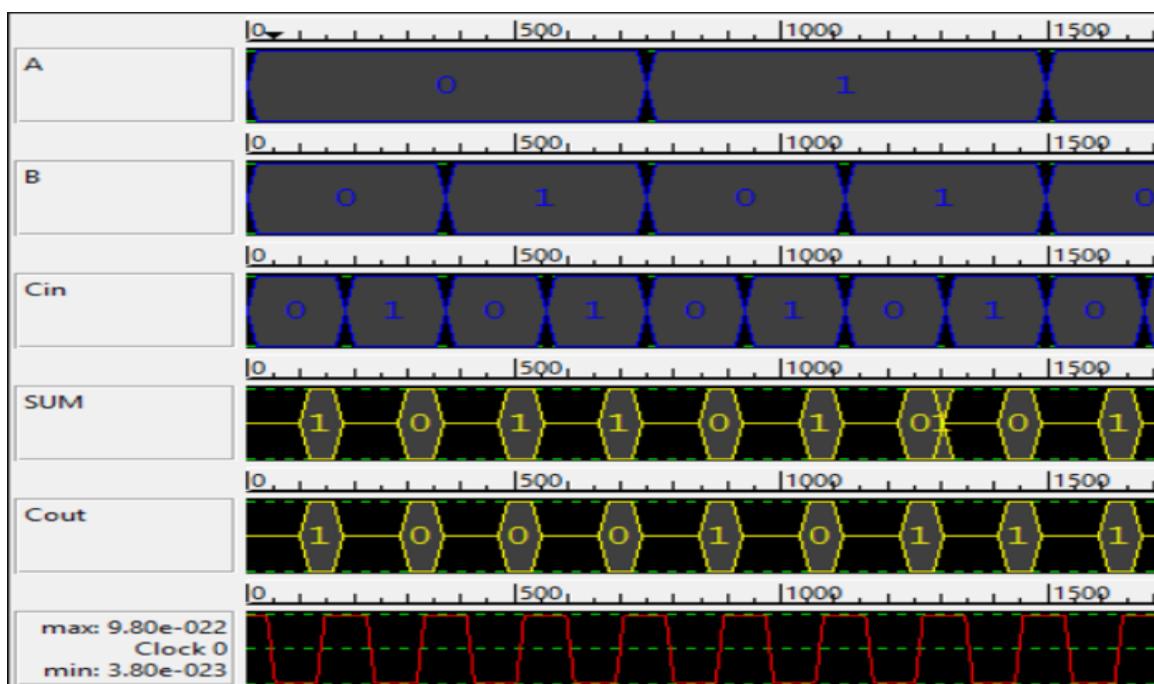


Figure 3.51: Simulation Result Adder Design 3

3.4.1.4 Design-4

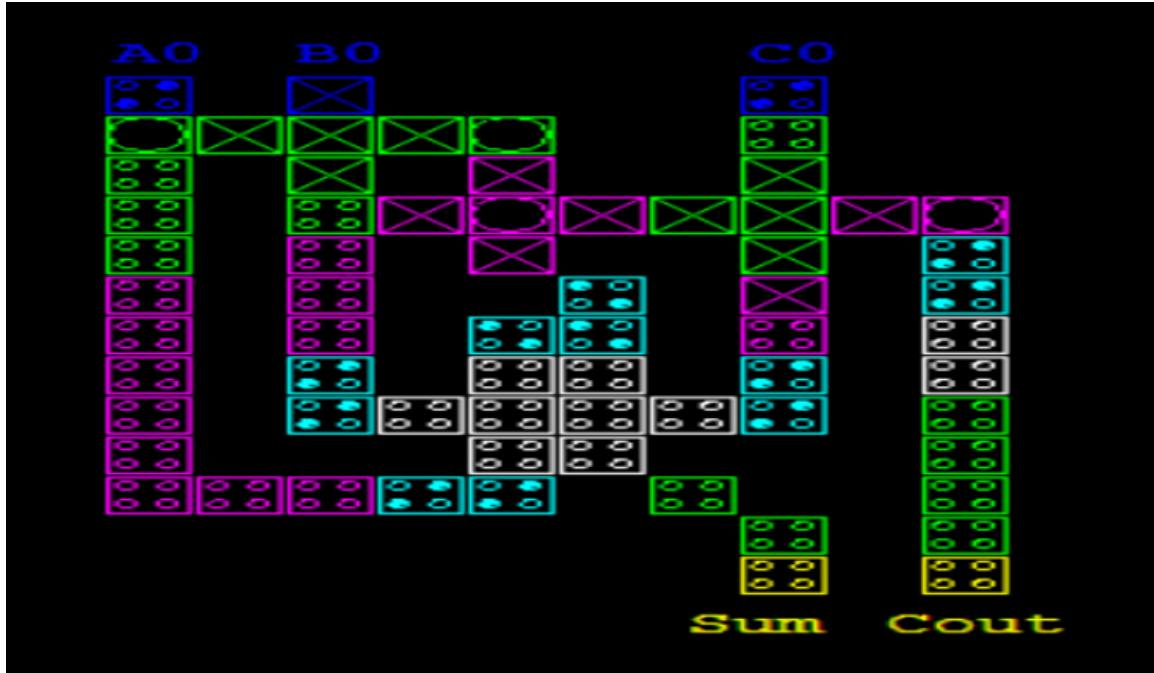


Figure 3.52: QCA Layout for Adder Design 4

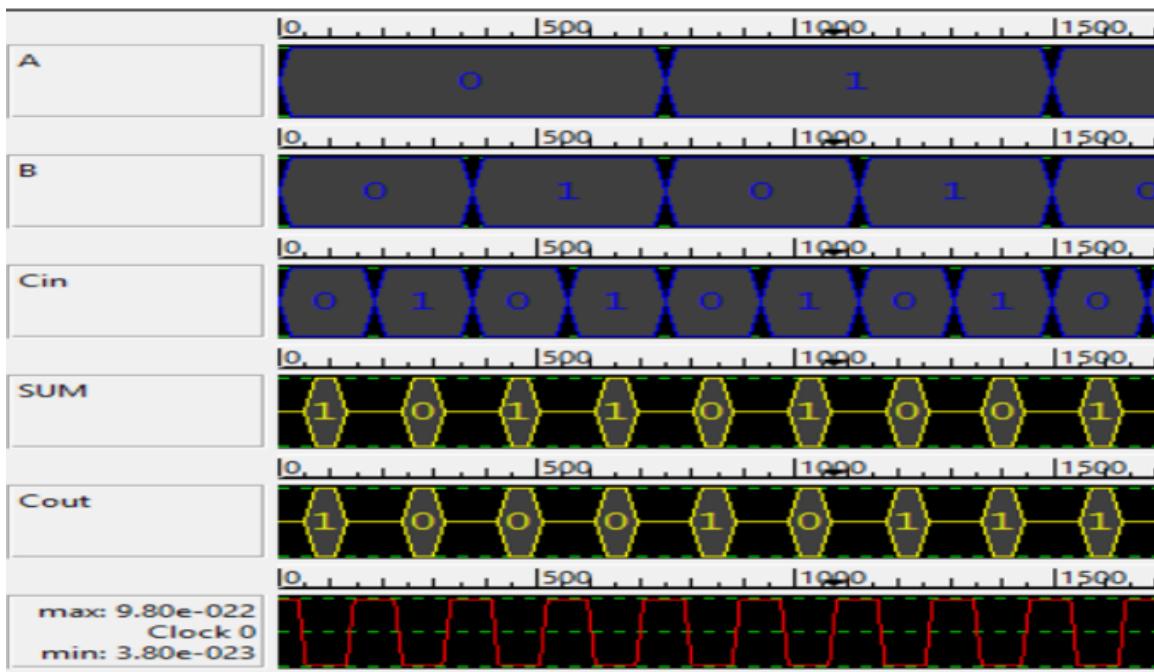


Figure 3.53: Simulation Result Adder Design 4

3.4.1.5 Comparison of 1-bit Adders

	Number of Cells	Area	Latency (Clock Cycles)	Simulation Time (Sec)	Crossover (Y/N)	Crossover Type
Design-1	118	0.12	2	69	Y	Multi-Layer
Design-2	88	0.08	1	54	Y	Multi-Layer
Design-3	83	0.03	1	56	N	N/A
Design-4	79	0.07	1	50	Y	Multi-Layer

Table 3.11: Comparative Study of 1-Bit Adders

3.4.2 2-Bit Adders

2-Bit Adders are also called as Ripple carry adders as they are used to add 2 two bit data input simultaneously. Generally Input provided are of two bits. The LSBs of both data (A_0 and B_0 as Fig. 3.35) are added first along with a carry from flag register. After this computation, MSBs are added along with the carry if generated from LSB addition and finally sum is computed by using outputs of both the full adders. Based upon the result, carry flag changes its state. The general procedure is as in Fig. 3.35

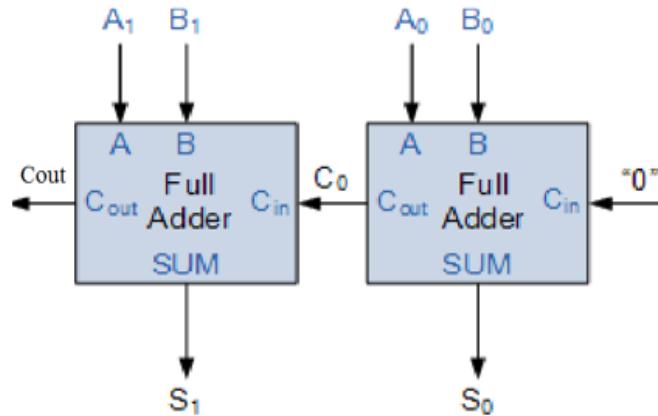
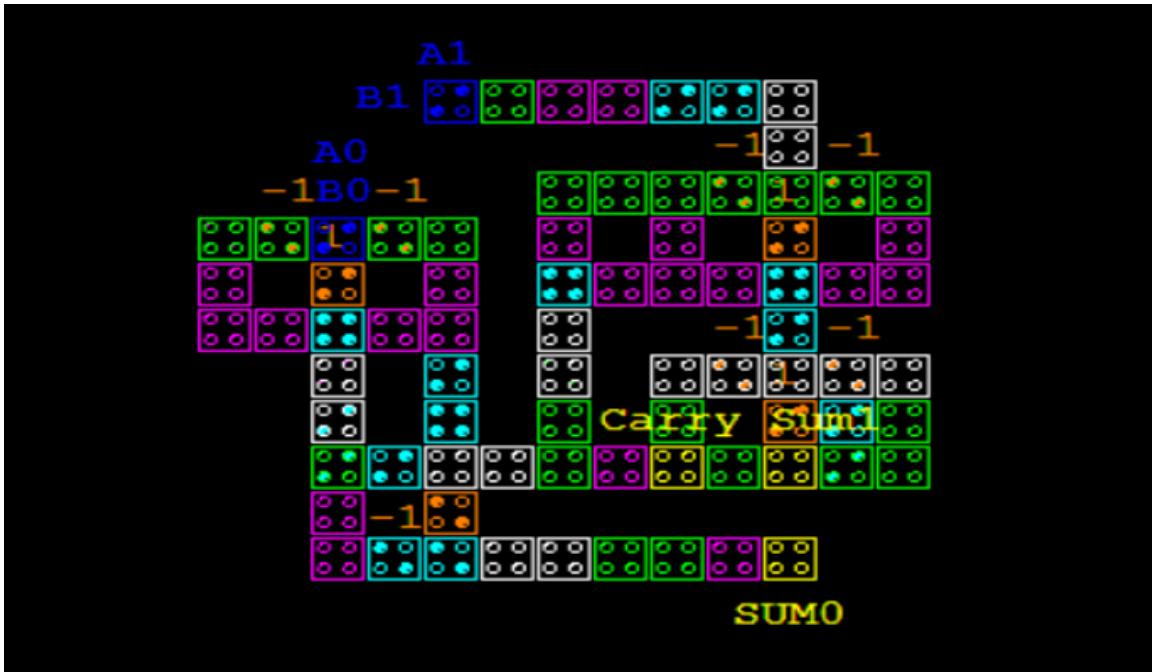
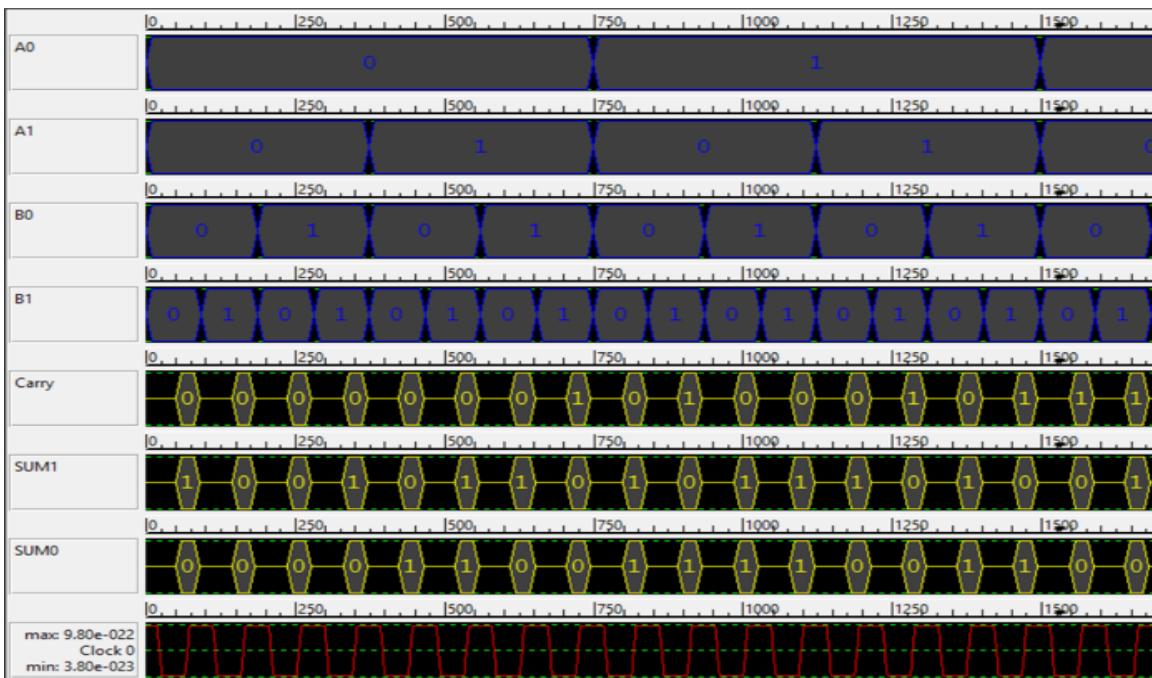


Figure 3.54: 2-bit Adder basic symbol

Previously, in 1 Bit adder section, Design 3 was made by using multi-layer approach. Using multi-layer approach is beneficial in terms of area and cell count because we can put one cell onto other in a specified manner. But we have to pay in terms of complexity and cost. As it is not easier to design multi-layers one onto other. By using fundamentals from 1-Bit adder we have implemented a 2 Bit adder in Multi-layer approach in QCA and verified by QCADesigner tool 2.0.3.



(a) QCA Layout



(b) Simulation Result

Figure 3.55: QCA Layout and Simulation Result for 2-Bit adder

3.4.3 4-Bit Adder

4-Bit adders are used to add two 4-bit numbers along with the carry C_{in} (taken from flag register as explained above). The general circuit for 4-bit adder is shown in Fig. 3.37. According to the Fig. two 4-bit numbers are firstly loaded into $A_3 - A_0$ and $B_3 - B_0$ (where $A_3 & B_3$ are MSBs and $A_0 & B_0$ are LSBs) and then taking carry flag status into account firstly applied to 1st full adder which computes addition for A_0, B_0 and C_{in} . The result of this adder is called S_0 (LSB of sum) and carry generated is fed to next stage as C_{in} . The same process continues until S_3 (MSB of sum) and C_{out} is generated[34].

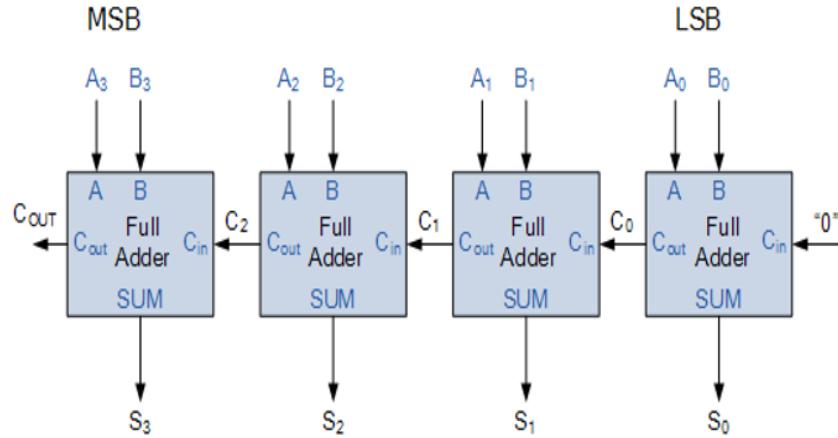


Figure 3.56: 4-bit Adder basic symbol

Stage	Previous Carry	Augends Bits (A)	Addend Bits (B)	SUM	Next Carry
0	$C_0 = 0$	$A_0 = 0$	$B_0 = 0$	$S_0 = 0$	$C_1 = 1$
1	$C_1 = 1$	$A_1 = 1$	$B_1 = 1$	$S_1 = 1$	$C_2 = 1$
2	$C_2 = 1$	$A_2 = 1$	$B_2 = 1$	$S_2 = 1$	$C_3 = 1$
3	$C_3 = 1$	$A_3 = 1$	$B_3 = 1$	$S_3 = 1$	$C_4 = 1$

Table 3.12: Truth Table for 4-bit adder ($A=0111$ and $B=1111$)

QCA Layout of 4-bit adder shown here uses a total of 442 cells with area of $0.74\mu m^2$. As clear from simulation result this design has a propagation delay of 4 clock pulses.

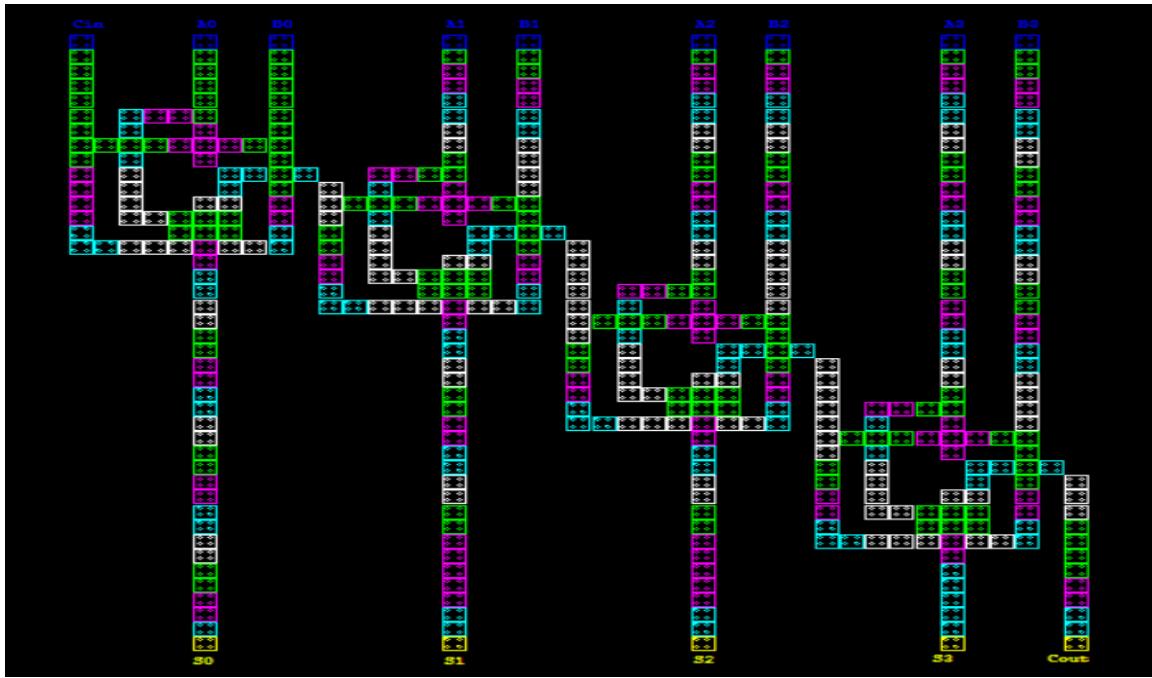


Figure 3.57: QCA Layout for 4-bit adder

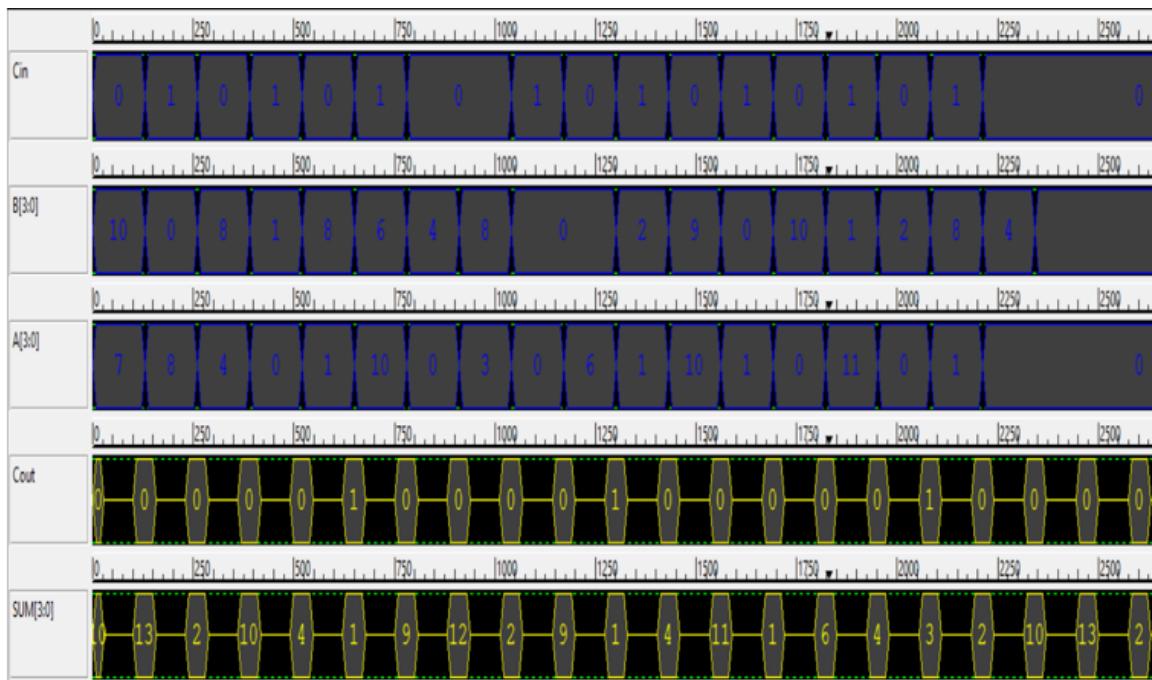


Figure 3.58: Simulation Result for 4-bit adder

Inputs	Active	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
Cin		<input checked="" type="checkbox"/>	0	1	0	1	0	1	0	0	1	0	1	0	1	0	1	0	1	0	0	0	0	0
Cin		<input checked="" type="checkbox"/>																						
B[3:0]		<input checked="" type="checkbox"/>	10	0	8	1	8	6	4	8	0	0	2	9	0	10	1	2	8	4	0	0	0	0
B3		<input checked="" type="checkbox"/>																						
B2		<input checked="" type="checkbox"/>																						
B1		<input checked="" type="checkbox"/>																						
B0		<input checked="" type="checkbox"/>																						
A[3:0]		<input checked="" type="checkbox"/>	7	8	4	0	1	10	0	3	0	6	1	10	1	0	11	0	1	0	0	0	0	0
A3		<input checked="" type="checkbox"/>																						
A2		<input checked="" type="checkbox"/>																						
A1		<input checked="" type="checkbox"/>																						
A0		<input checked="" type="checkbox"/>																						

Figure 3.59: Simulation vector for 4-bit adder

3.5 Multiplier Designing

Along with adders, Multipliers are equally important in arithmetic and logical operations. In this section we have implemented a 2-bit multiplier which uses the following procedure:

X_1	X_0	Y_1	Y_0	Z_3	Z_2	Z_1	Z_0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0
0	0	1	0	0	0	0	0
0	0	1	1	0	0	0	0
0	1	0	0	0	0	0	0
0	1	0	1	0	0	0	1
0	1	1	0	0	0	1	0
0	1	1	1	0	0	1	1
1	0	0	0	0	0	0	0
1	0	0	1	0	0	1	0
1	0	1	0	0	1	0	0
1	0	1	1	0	1	1	0
1	1	0	0	0	0	1	1
1	1	1	0	0	1	1	0
1	1	1	1	1	0	0	1

Table 3.13: Truth Table for 2-Bit Multiplier

Truth table for 2 bit multiplier is shown above having $X_1 X_0$ & $Y_1 Y_0$ as 2-bit inputs and $Z_3 Z_2 Z_1 Z_0$ as output. After analysing this table, we get following equations

$$Z_0 = X_0 \cdot Y_0$$

$$Z_1 = X_0 Y_1 \oplus X_1 Y_0$$

$$Z_2 = X_1 \cdot Y_1 \cdot \overline{(X_0 \cdot Y_0)}$$

$$Z_3 = X_0 \cdot Y_0 \cdot X_1 \cdot Y_1$$

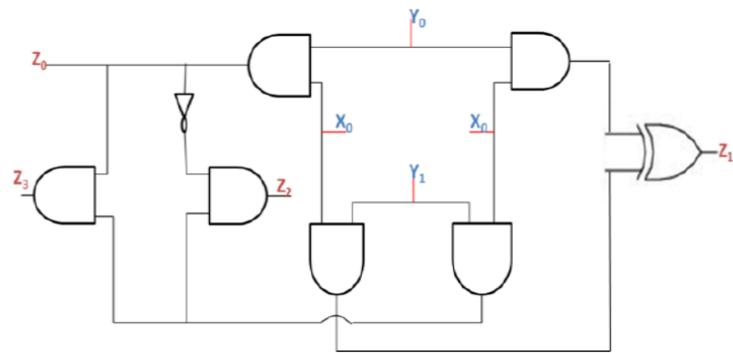


Figure 3.60: Schematic Circuit for Multiplier

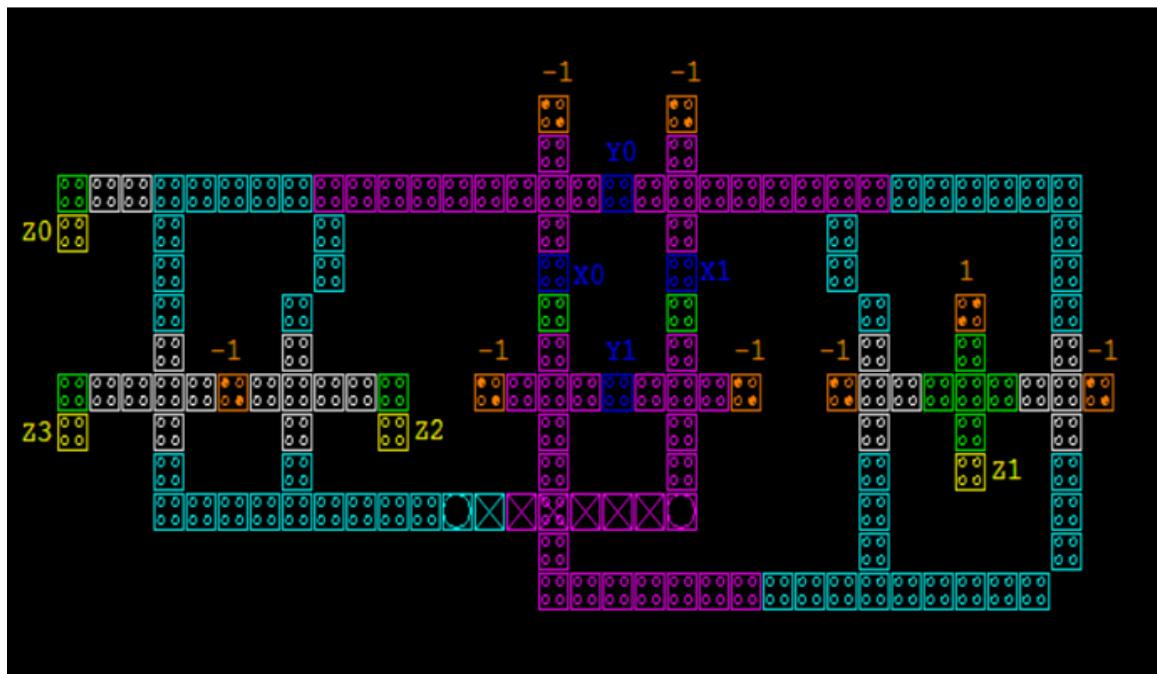


Figure 3.61: QCA Layout for 4-bit adder

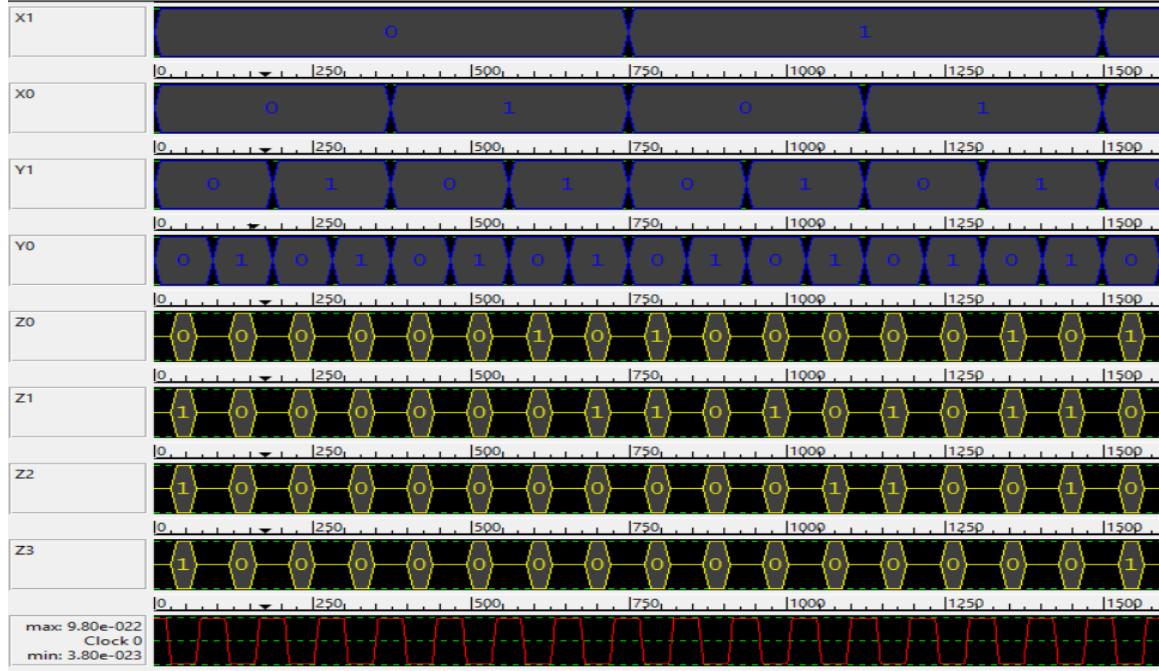


Figure 3.62: Simulation Vector for 4-bit adder

Chapter 4

Designing and Implementation of Random Access Memory (RAM) in QCA

4.1 Introduction

Random Access Memory (RAM) can be considered as a short term memory which holds information as long as supply is on. Hence it falls under volatile memory category. Designing of RAM in QCA generally done in one of two ways: Loop based or Line based, suitable to the operational manner. In loop based structures, a loop consisting all four clocking zones is created to store the information. While in line based structures, a QCA line is used to save the output.

One of the basic structure used in RAM designing is D Latch. As shown in Fig. 4.1, it is clear that if Enable input is zero then it causes Input to be zero and the value does not change inside the loop. While if Enable is “1” then value in the loop changes to input value.

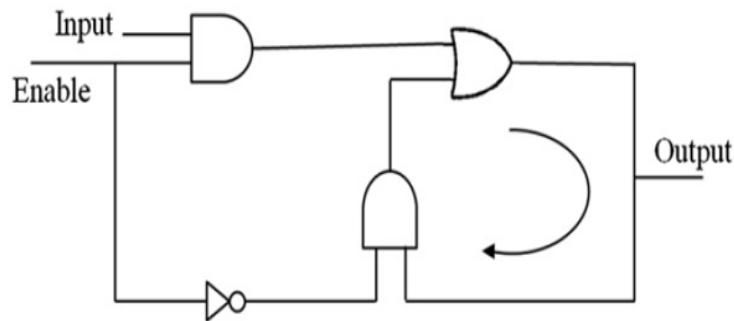


Figure 4.1: D Latch Circuit Diagram

4.2 Basic RAM Design

Each RAM cell has two inputs and one output. If $Write/\overline{Read}$ input is high (“1”), input path is activated and data input is written to it. And when this signal goes low i.e. “0”, the data can only be

read via output path. In order to design a RAM cell, firstly a new D-Latch structure is designed shown in Fig. 4.2. If Enable input is low, the value of ‘a’ is equals to Q and value of ‘b’ is also equals to Q. In this manner the inputs to Majority gate will be Q, Q and Input(D). Hence the output of MV will be Q.

Now, if Enable is high (“1”), then values of ‘a’ and ‘b’ are ‘1’ & ‘0’ respectively. And input to MV are 1, 0 & Input(D) which leads MV to give output same as input[28].

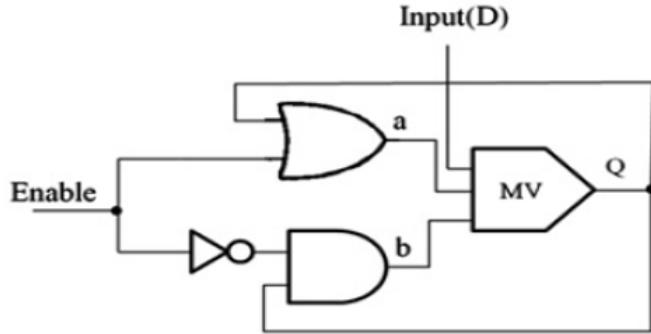


Figure 4.2: D Latch Circuit Diagram

Input	Enable	$Q(t)$	a	b	$Q(t+1)$
0	0	0	0	0	0
0	0	1	1	1	1
0	1	0	0	0	0
0	1	1	1	1	1
1	0	0	1	0	0
1	0	1	1	0	0
1	1	0	1	0	1
1	1	1	1	0	1

Table 4.1: Truth Table for D Latch

In order to design RAM cell, we must ensure that when $\overline{Write/Read}$ signal is low, $Q(t+1)$ should be equals to $Q(t)$, so does output specifies a read operation. And when this signal is high, output should equals to “0” and $Q(t+1)$ equals to Input, specifies write operation. Such RAM structure is shown in Fig. 4.3.

If we observe Table 4.2, we find that it is as same as D latch table where Enable and ‘b’ are replaced by Write and Output respectively. This table satisfies above said conditions and hence a QCA Layout is designed and simulated[28].

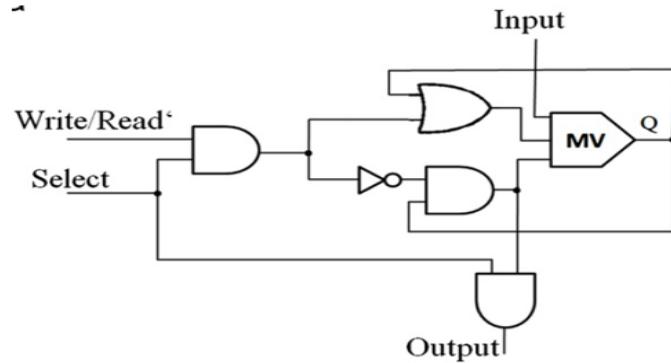


Figure 4.3: Schematic Diagram for BAsic RAM Cell

Write	Input	$Q(t)$	a	Output	$Q(t+1)$
0	0	0	0	0	0
0	0	1	1	1	1
0	1	0	0	0	0
0	1	1	1	1	1
1	0	0	1	0	0
1	0	1	1	0	0
1	1	0	1	0	1
1	1	1	1	0	1

Table 4.2: Truth Table for D Latch

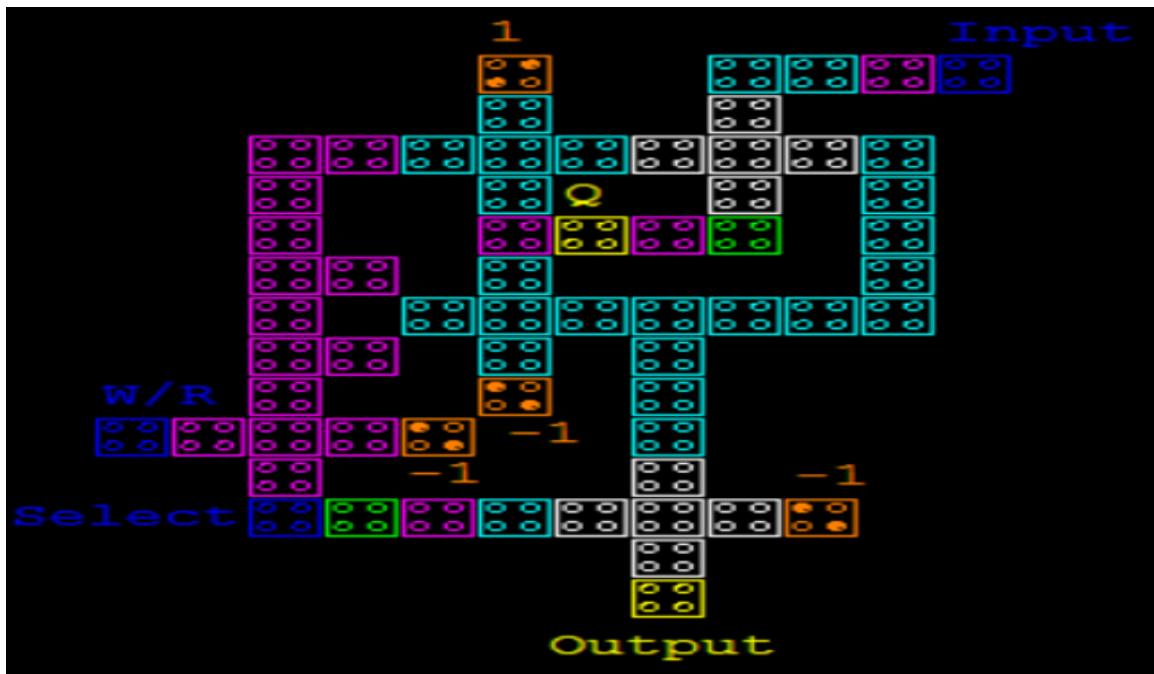


Figure 4.4: QCA Layout for BAsic RAM Cell

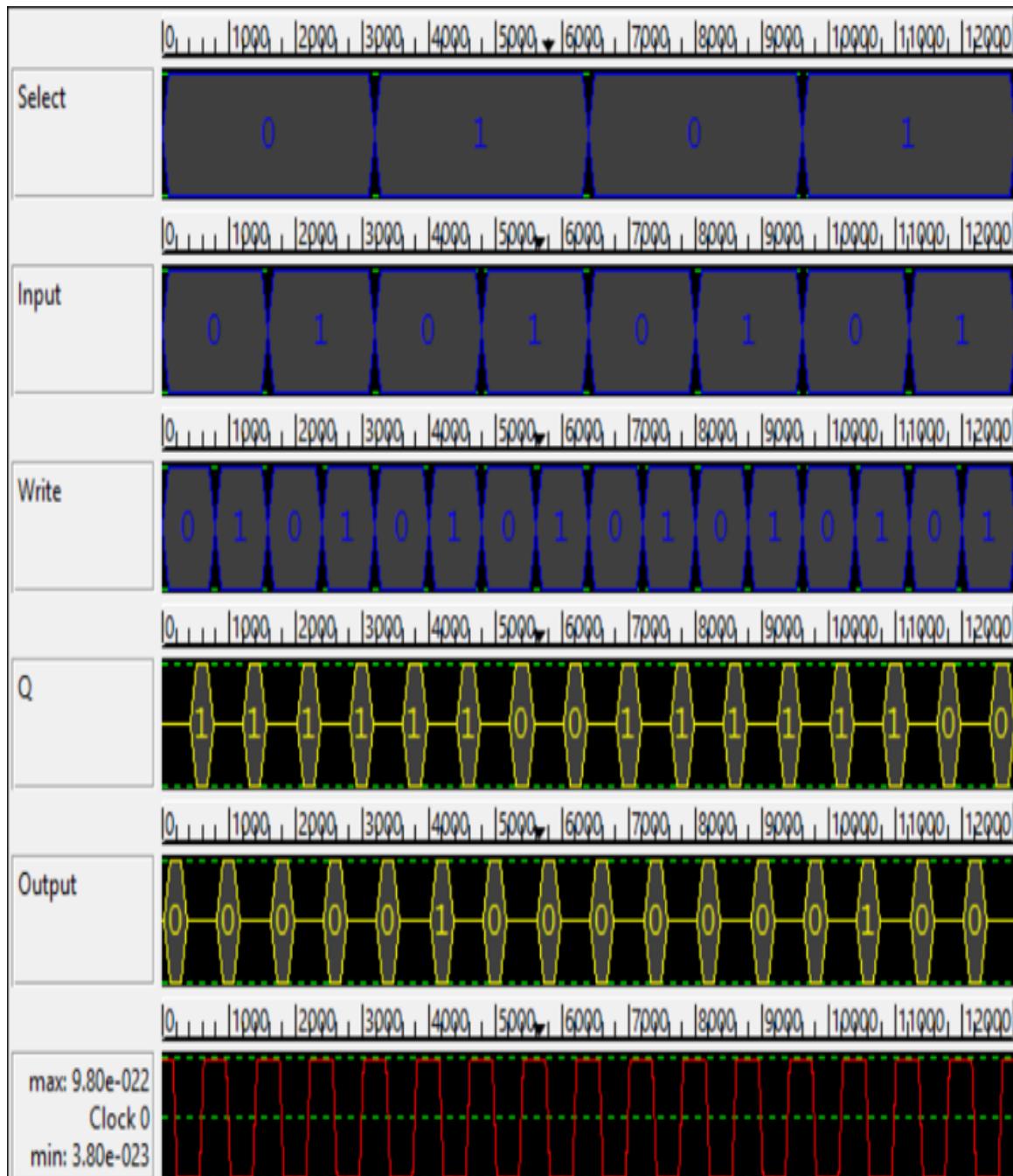


Figure 4.5: Simulation Result for Basic RAM Cell

4.3 RAM Cell with Set and Reset Switches

A RAM cell with Set/Reset capability is shown in Fig. 4.6. As per the Fig. 4.6, it consists of three 3-input MV gates and one 5-input MV gate. Set and Reset work as controlled inputs. During normal operation ($Set = 0$ & $Reset = 1$) with *Select* and *WRITE* signal as high, data present at input is transmitted to output which specifies an write operation. Under same condition except *WRITE* = 0, the read operation accomplished. Under the Set mode, the output of RAM cell is set to “1” with Set and Reset signal high. Similarly, Reset mode is achieved by setting both Set and Reset inputs to “0” and hence output of cell is set to “0” irrespective of the state of any other signal[10].

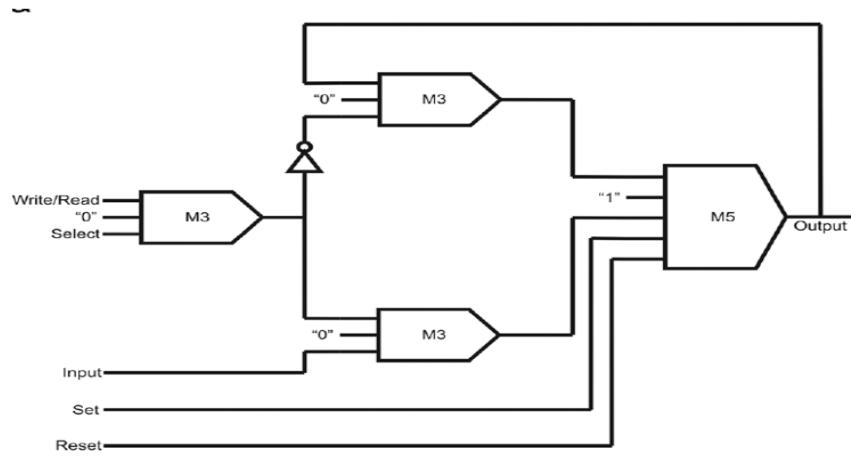


Figure 4.6: Schematic Diagram for RAM Cell with Set and Reset Switches

Mode of Operation	Type of Operation	Select	Set	Reset	<i>Write/Read</i>	Input	Previous Outputs	Output
Normal	Write	1	0	1	1	1	x	1
	Write	1	0	1	1	0	x	0
Normal	Read	1	0	1	0	x	0	0
	Read	1	0	1	0	x	1	1
Set	Set	x	1	1	x	x	x	1
Reset	Reset	x	0	0	x	x	x	0

Table 4.3: Truth Table for RAM Cell

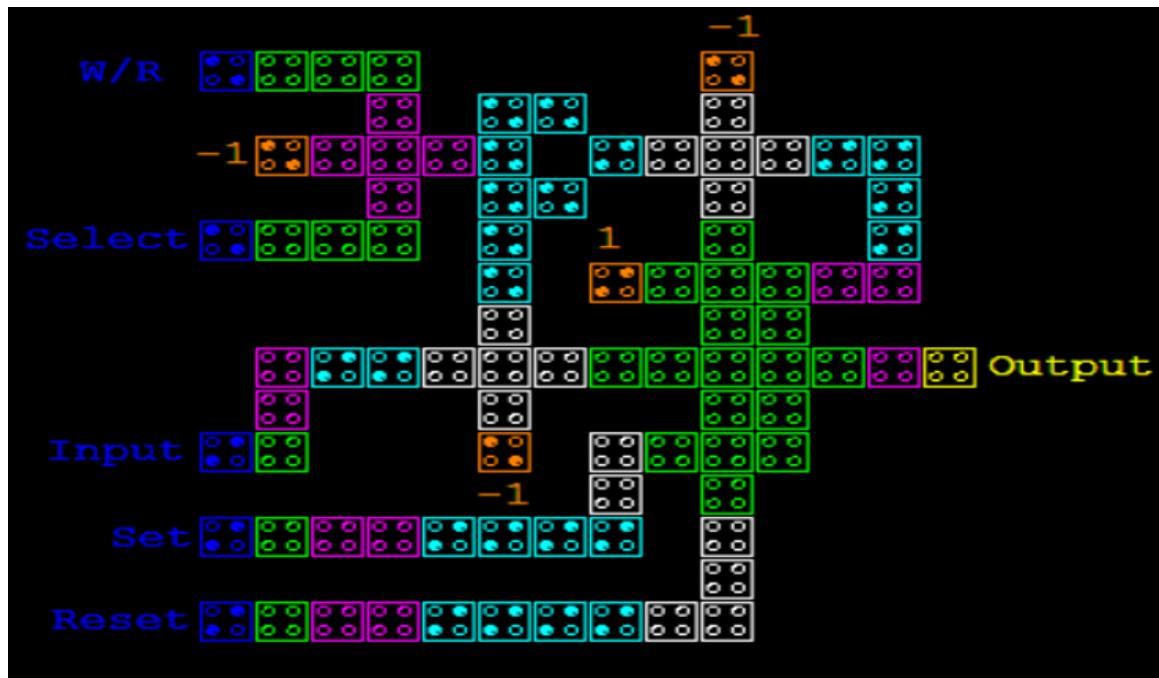


Figure 4.7: QCA Layout for RAM Cell with Set and Reset Switches

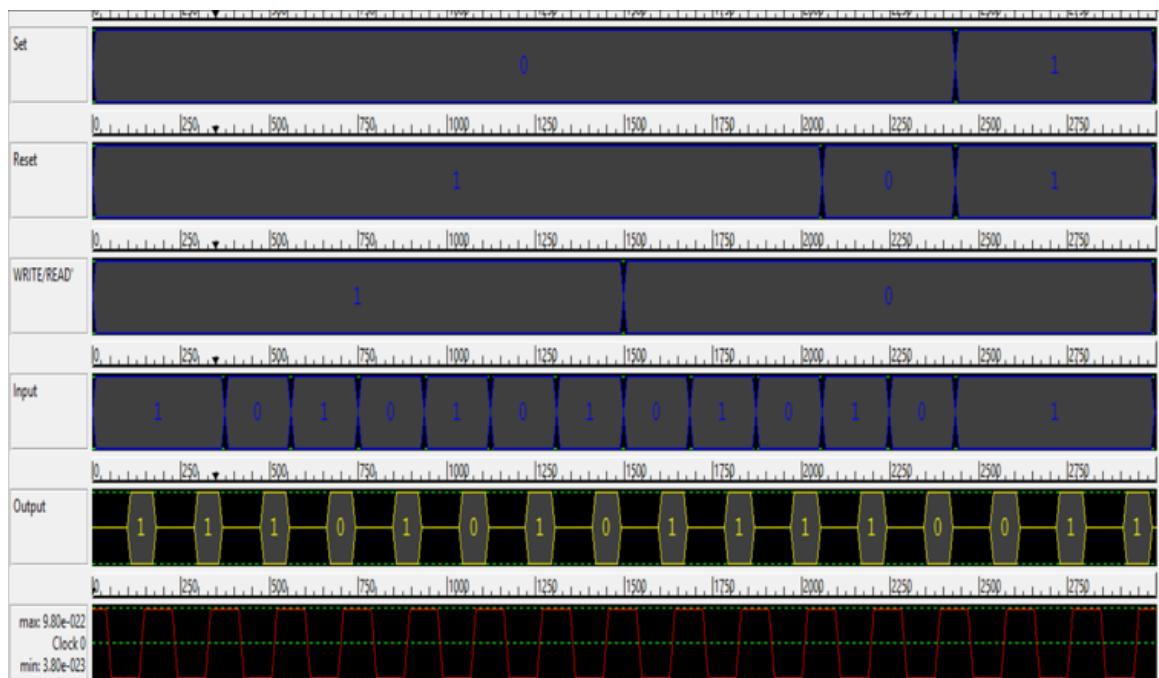


Figure 4.8: Simulation Result for RAM Cell with Set and Reset Switches

4.4 Advance RAM Structure

Although, previous design has set and reset abilities but it uses different switches for each logic, hence increases cell area. The schematic diagram to implement RAM cell is shown in Fig. 4.9[22].

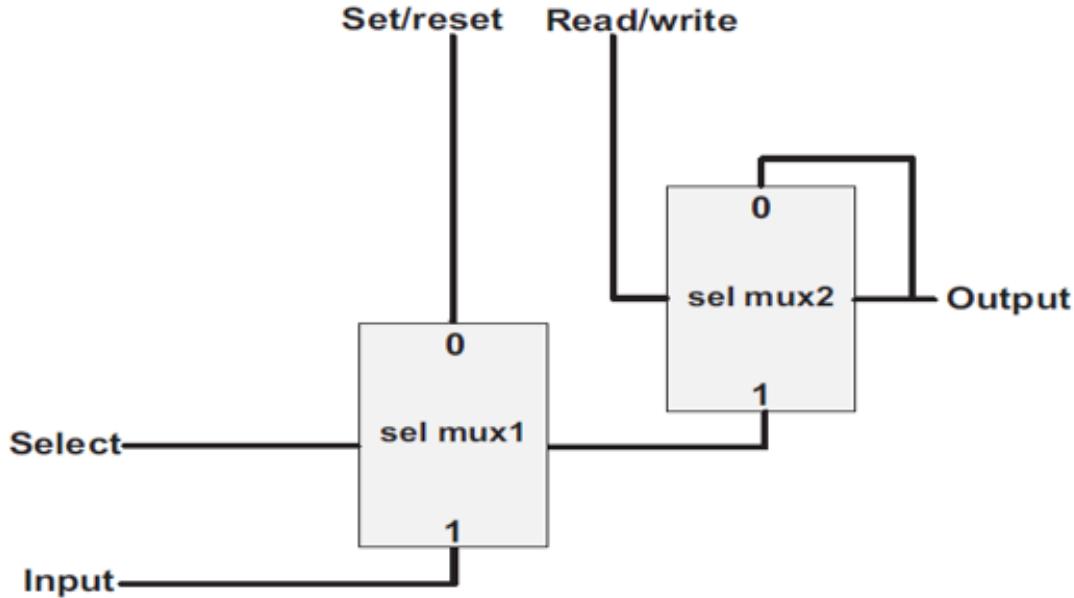


Figure 4.9: Schematic Diagram for advance RAM Cell

As per Fig. 4.9, Input to MUX 1 are set/reset and input data and select line is used to select appropriate input to be driven to output. This output along with output of MUX2 are given as input to MUX 2. Here *Write/Read* signal works as select line and selects the required value. Operation of this circuit is explained in Table 4.4. According to the table, when *Write/Read* signal is “0” the output does not change whereas when it goes high “1”, either input or set/reset value can be directed to the output. In such case, select line role of MUX1 becomes significant. If it is “0”, set/reset is transmitted otherwise input data is transmitted to output[22].

<i>Write/Read</i>	Select	Set/Reset	Out (t)
0	x	x	Out (t-1)
1	0	0	0
1	0	1	1
1	1	x	Input

Table 4.4: Truth Table for Advance RAM Circuit

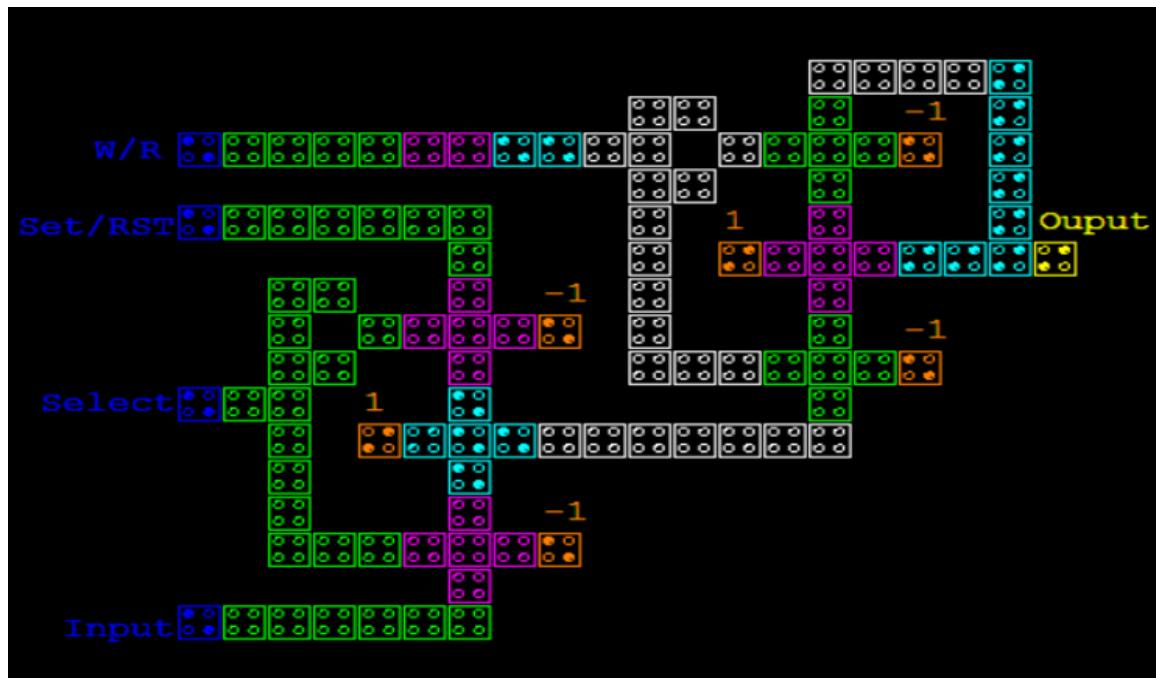


Figure 4.10: QCA Layout for advance RAM Cell

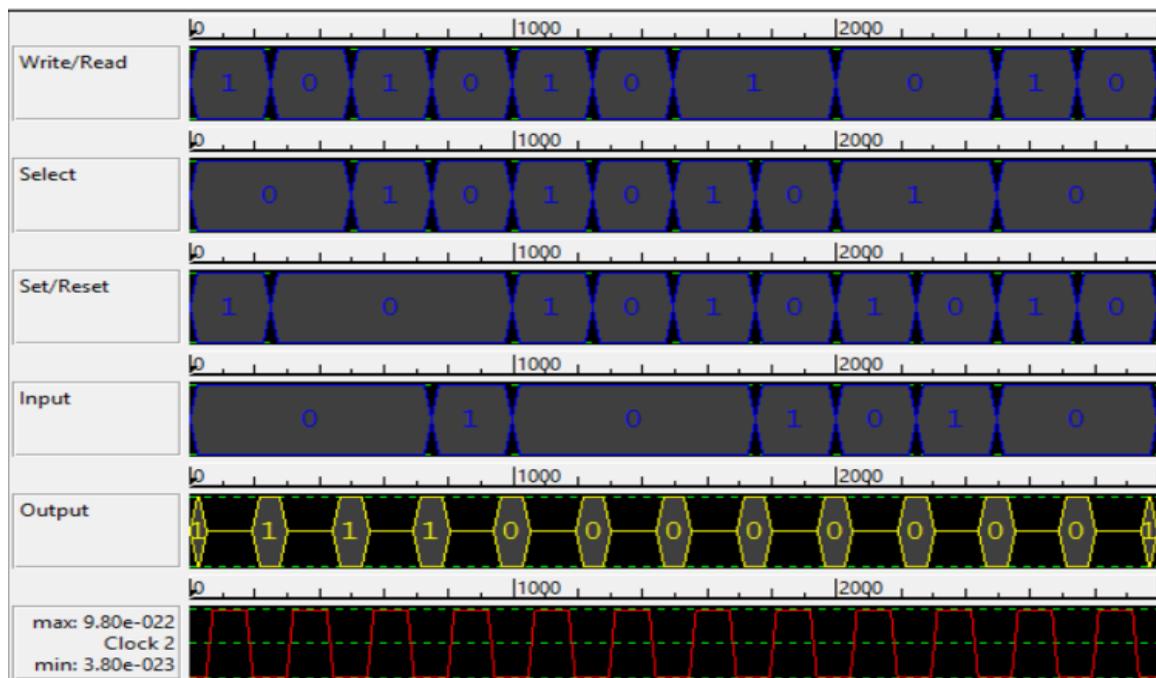


Figure 4.11: Simulation Result for advance RAM Cell

Chapter 5

Designing and Implementation of Reversible Gates in QCA

5.1 Introduction

As we increase density of logic elements and clock them at higher frequencies, higher will be the heat dissipation. This creates at least three problems[9]:

- Energy costs money
- Systems overheat
- Portable systems exhaust their batteries

When a computational system erases a bit of information, it must dissipate $\ln 2 \times kT$ energy, where k is Boltzmann's constant and T is the temperature. For $T = 300$ Kelvins (room temperature), this is about 2.9×10^{-21} joules. This is roughly the kinetic energy of a single air molecule at room temperature.

Whenever an operation is computed, a logic bit is erased. These operations are therefore called as irreversible operations. These erasure is inefficient and dissipates more energy. But our requirement is to reduce the power dissipation dissipated by each operation. We can use two approach for above said problem as to increase the efficiency of erasure system or using reversible operations, which uses less than specified limit of energy.

What Reversibility in computing implies about computational states is that information can never be lost. Hence we can recover any earlier stage by computing backward or un-computing the results. But to gain the benefits of logical reversibility, firstly we need to perform Physical reversibility. Physical reversibility is a process that dissipates no energy to heat. But practically it is not possible to achieve perfect physical reversibility. When voltage levels change from (+) to (-) in computing systems: bits from 0 to 1. The energy that is needed to make that change is dissipated in the form of heat. Rather than changing voltages to new levels, reversible circuit elements do not change voltage levels, they will gradually move charge from one node to the next. This way, only a minute amount of energy on each transition is lost.

5.2 Basic Terms of Reversible Logic

Here are some of the basic terms related to reversible logic are explained[24]:

1. **Reversible Function** : If the number of outputs is identical to a number of inputs for a Boolean function $f(x_1, x_2, \dots, x_n)$ with multiple output of n Boolean variables, it is known as reversible function. Any output pattern has a unique pre-image which specifies a one to one mapping.
2. **Reversible Logic** : Each output has a particular set for particular input set.
3. **Garbage Output** : In order To maintain one to one mapping between input and output, the number of extra outputs added to make an n-input-k-output function ((n;k)function) reversible is called garbage output. The relation between number of garbage outputs and constant inputs is as below:

$$Input + constant\ input = output + garbage$$

Garbage line is necessary to sustain the reversibility.

4. **Quantum Cost** : The amount of work done to transform a reversible circuit to a quantum circuit. It is easy to calculate if one knows the number of primitive reversible logic gates needed to identify the circuit.
5. **Ancilla inputs** : These are constant inputs and remain constant to incorporate the given logical function.
6. **Flexibility** : Flexibility is the universality of a reversible logic gate for realizing more functions.
7. **Hardware complexity** : It describes the number of logic actions.

5.3 Reversible Logic Gates

It is an n-input n-output logic device with one-to-one mapping. It means there is a unique way to recover the outputs from the inputs and vice versa. Fan-Out is not allowed in the synthesis process of reversible gates as one-to-many concept is not reversible. But using additional gates one can achieve fan out in reversible circuits. Minimum number of reversible logic gates should be used to design such circuits. There can be many parameters for determining the performance and complexity of circuits. Reversible logic gates can also be implemented on QCA Designer tool. Implementation and simulation of reversible logic gates along with their circuit diagram and tables are shown in next section[19][24][39].

5.3.1 CNOT Gate

It is a 2*2 reversible logic gate. It is also known as controlled NOT gate.

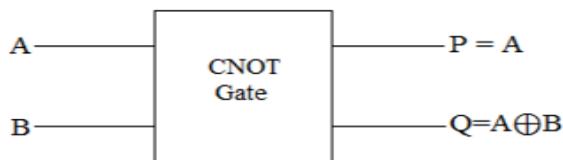


Figure 5.1: Symbol for CNOT Gate

A	B	P	Q
0	0	0	0
0	1	0	1
1	0	1	1
1	1	1	0

Table 5.1: Truth Table for CNOT Gate

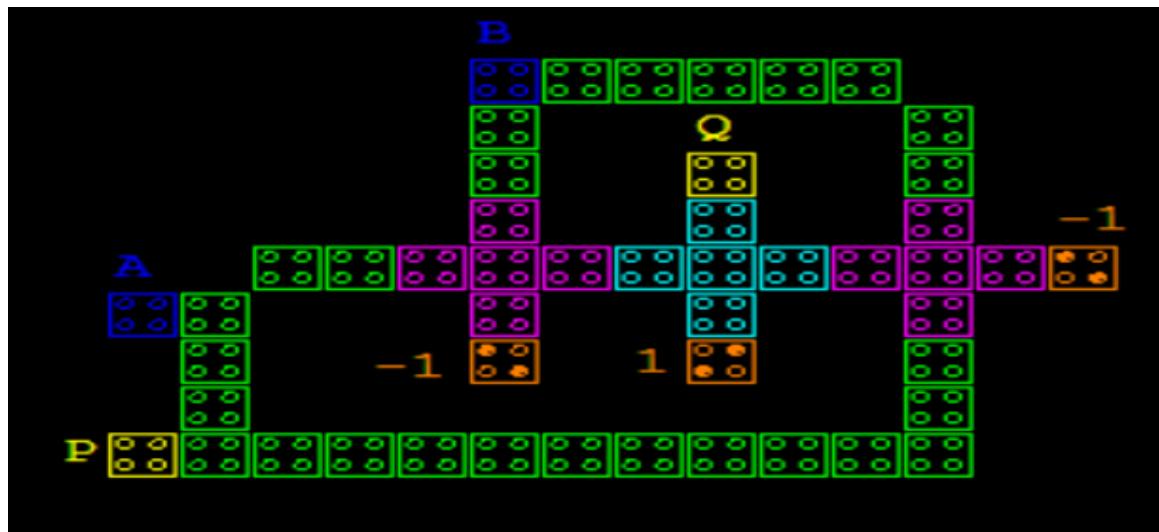


Figure 5.2: QCA Layout of CNOT Gate

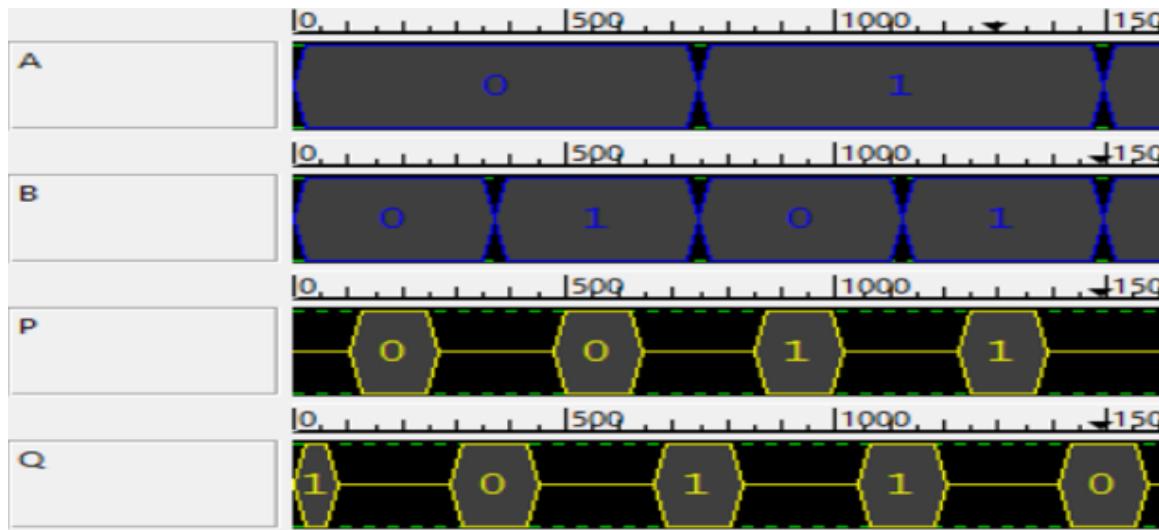


Figure 5.3: Simulation Result of CNOT Gate

5.3.2 FEYNMAN Gate

It is also a 2×2 gate and widely used for a Fan-Out purpose.

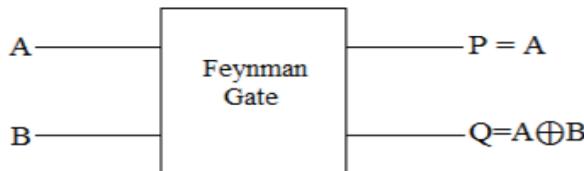


Figure 5.4: Symbol for FEYNMAN Gate

A	B	P	Q
0	0	0	0
0	1	0	1
1	0	1	1
1	1	1	0

Table 5.2: Truth Table for FEYNMAN Gate

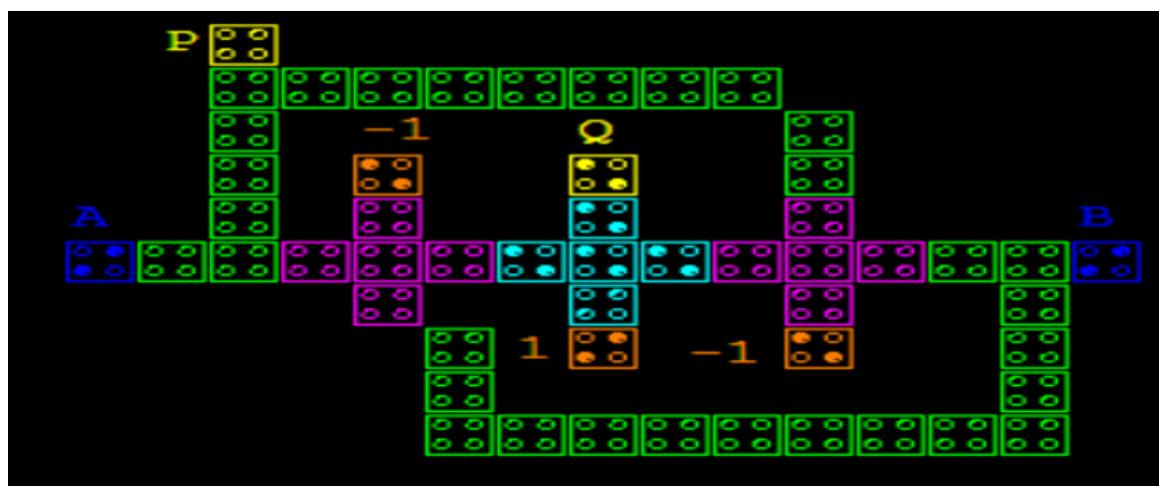


Figure 5.5: QCA Layout of FEYNMAN Gate

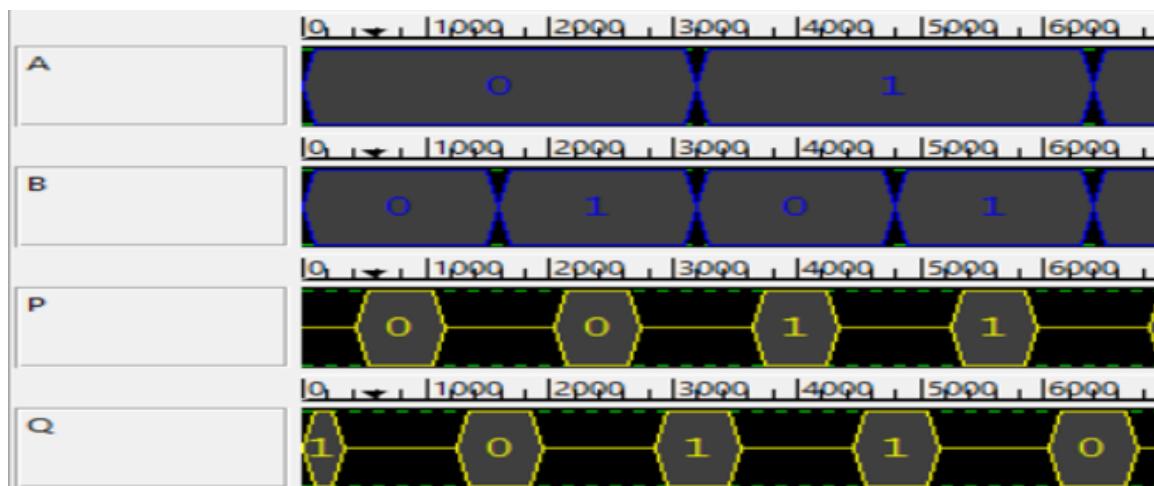


Figure 5.6: Simulation Result of FEYNMAN Gate

5.3.3 Double FEYNMAN Gate

This is a modification to FEYNMAN gate. This is a three input & three output gate. At the output side, we have one of the inputs i.e. A and logical XOR of A with B & C respectively.

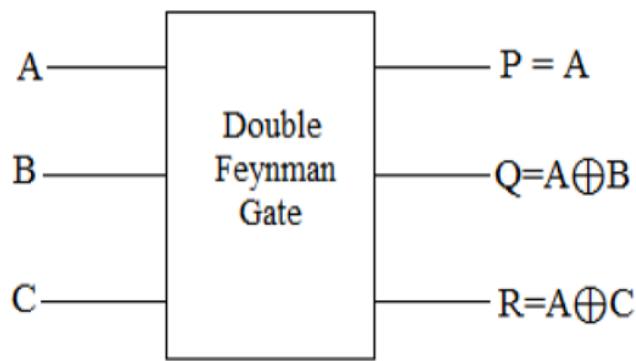


Figure 5.7: Symbol for Double FEYNMAN
Gate

A	B	C	P	Q	R
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	0
0	1	1	0	1	1
1	0	0	1	1	1
1	0	1	1	1	0
1	1	0	1	0	1
1	1	1	1	0	0

Table 5.3: Truth Table for Double FEYNMAN Gate

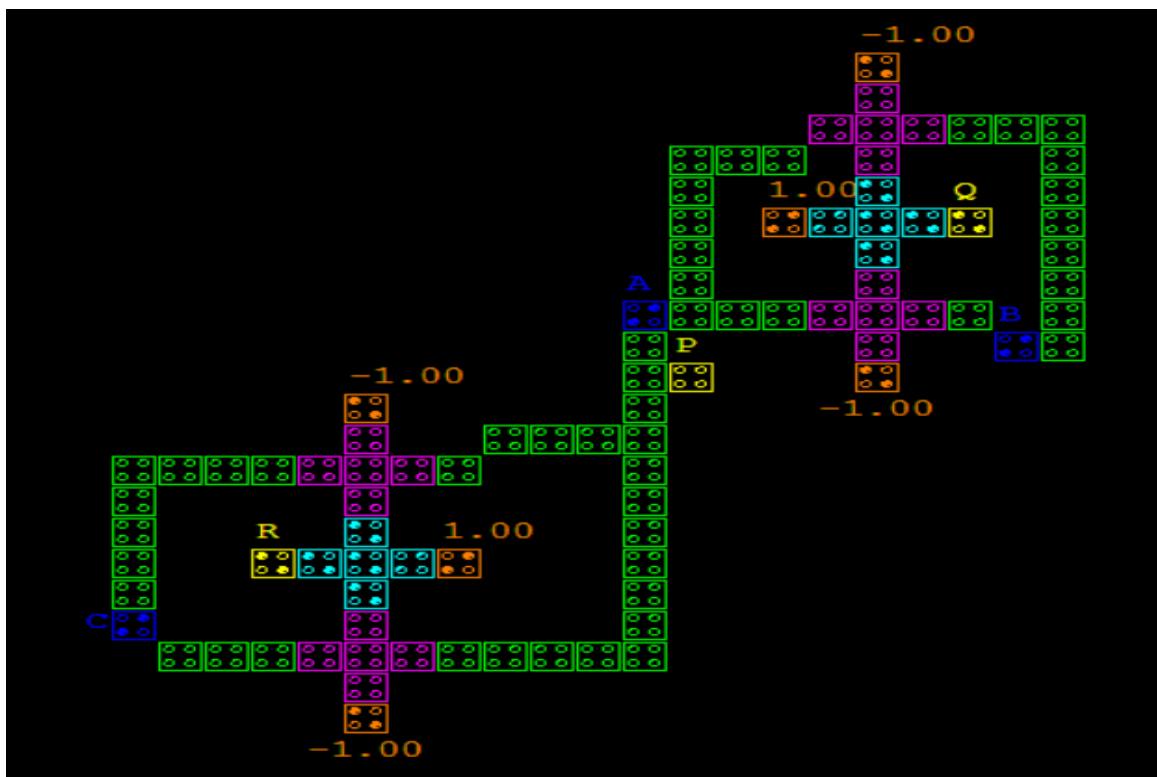


Figure 5.8: QCA Layout of Double FEYNMAN Gate

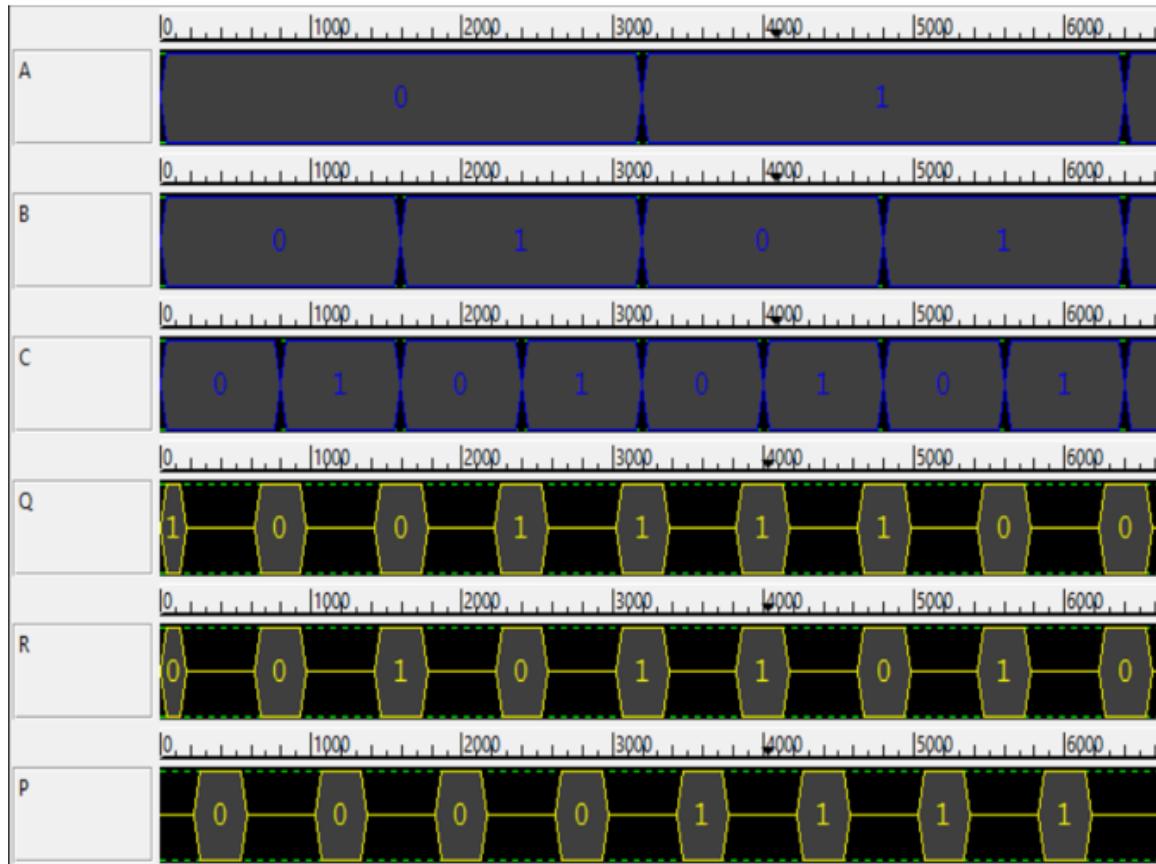


Figure 5.9: Simulation Result of Double FEYNMAN Gate

5.3.4 TOFFOLI Gate

This is 3×3 reversible logic gate. Basic symbol, truth table, and QCA implementation are shown below.

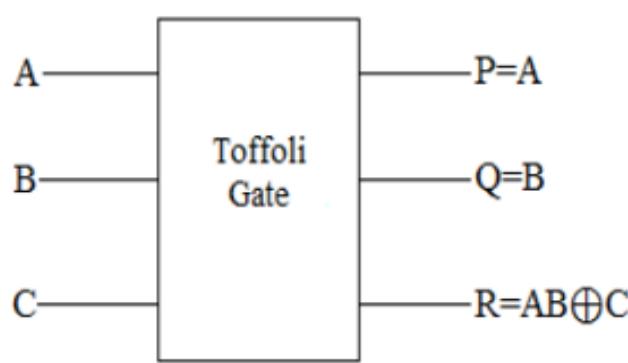


Figure 5.10: Symbol for TOFFOLI Gate

A	B	C	P	Q	R
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	0
0	1	1	0	1	1
1	0	0	1	0	0
1	0	1	1	0	1
1	1	0	1	1	1
1	1	1	1	1	0

Table 5.4: Truth Table for TOFFOLI Gate

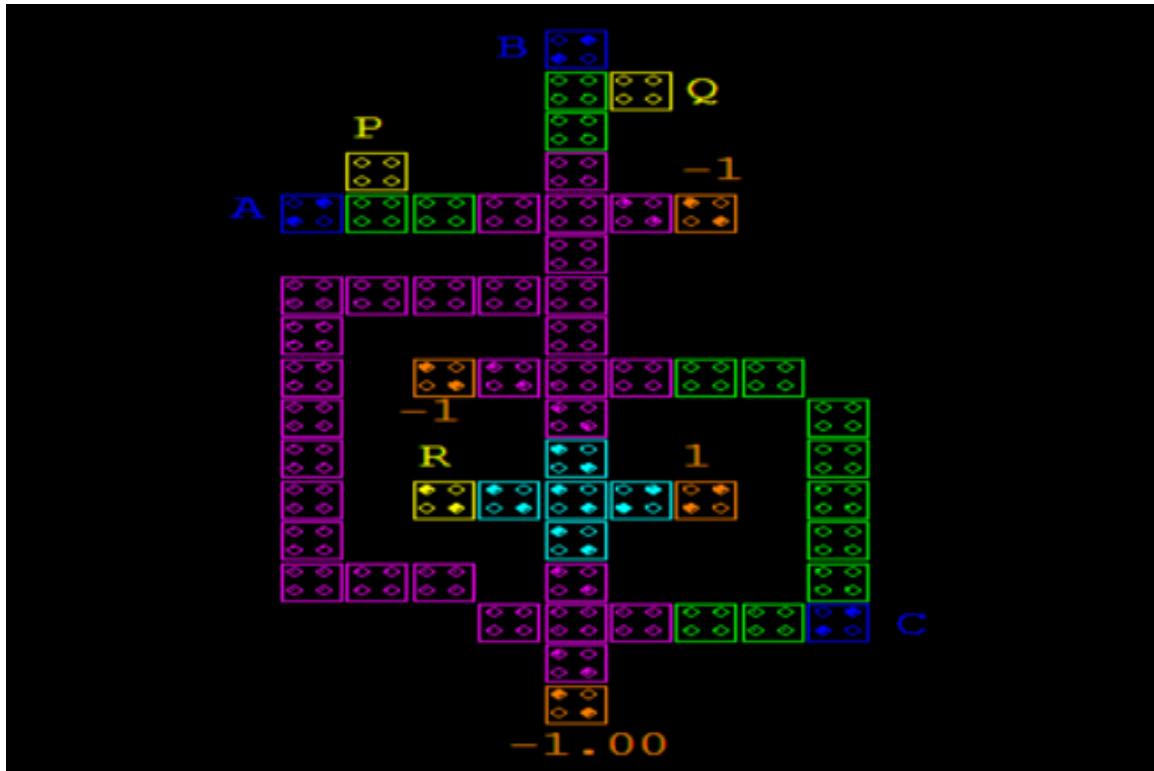


Figure 5.11: QCA Layout of TOFFOLI Gate

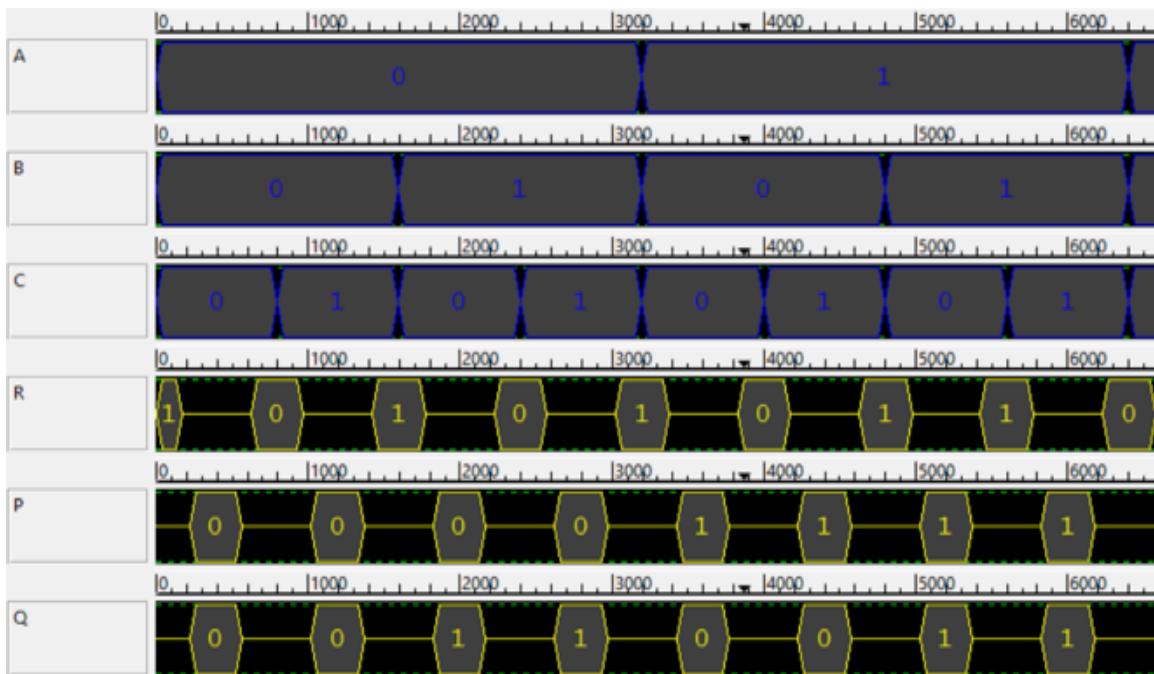


Figure 5.12: Simulation Result of TOFFOLI Gate

5.3.5 FREDKIN Gate

This is also a three input & three output reversible gate. It provides outputs as

$$P = A$$

$$Q = A'B + AC$$

$$R = AB + A'C$$

Due to these outputs, it uses more number of primitive gates which contributes to its quantum cost. Hence this gate has a higher quantum cost.

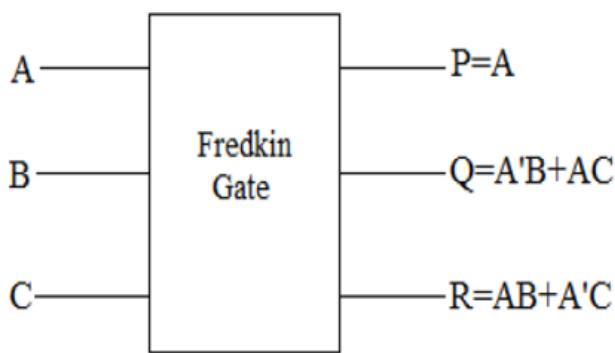


Figure 5.13: Symbol for FREDKIN Gate

A	B	C	P	Q	R
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	0
0	1	1	0	1	1
1	0	0	1	0	0
1	0	1	1	1	0
1	1	0	1	0	1
1	1	1	1	1	1

Table 5.5: Truth Table for FREDKIN Gate

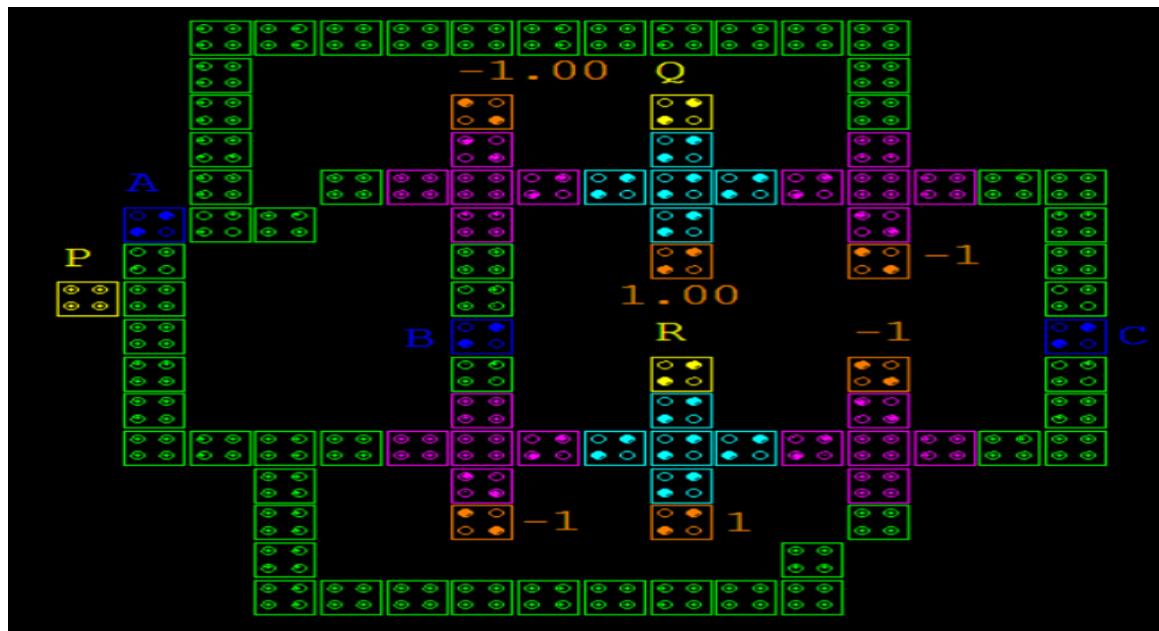


Figure 5.14: QCA Layout of FREDKIN Gate

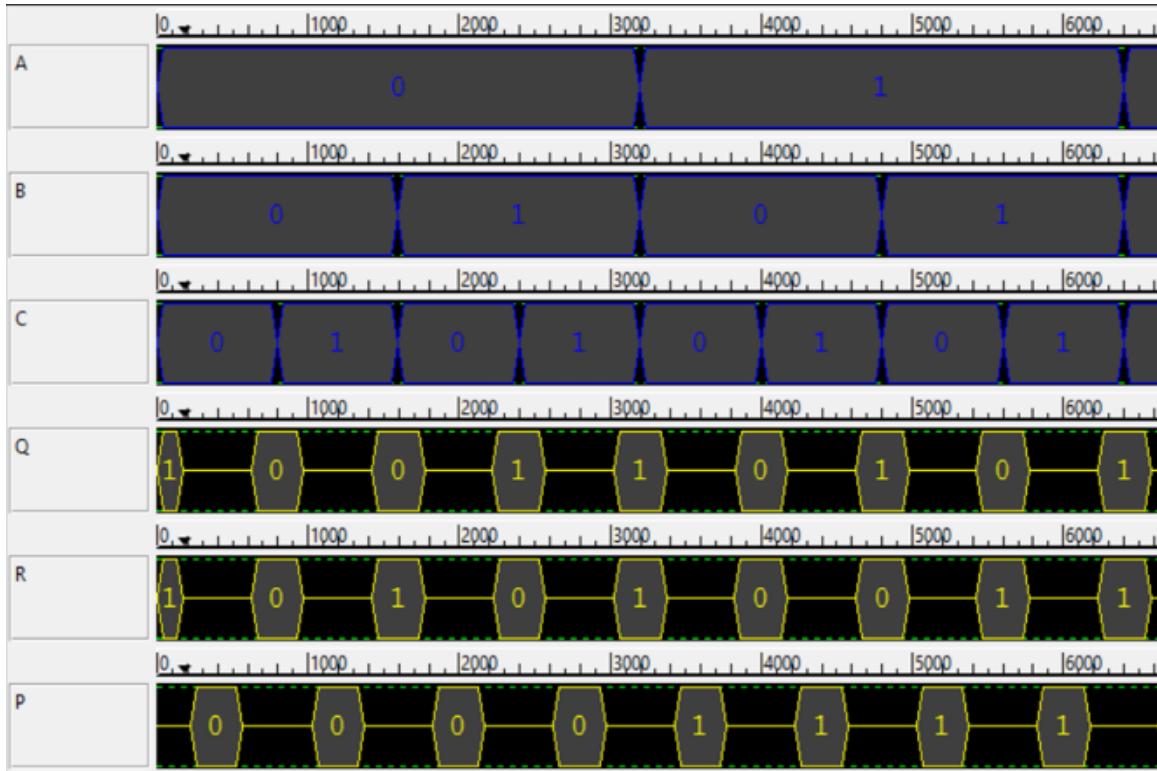


Figure 5.15: Simulation Result of FREDKIN Gate

5.3.6 BJT Gate

This is an 3-input, 3-output reversible gate. This provides two of the inputs [A & B] at the output directly. And provides logical XOR of third input C with product of logical OR operation of two inputs i.e. A & B.

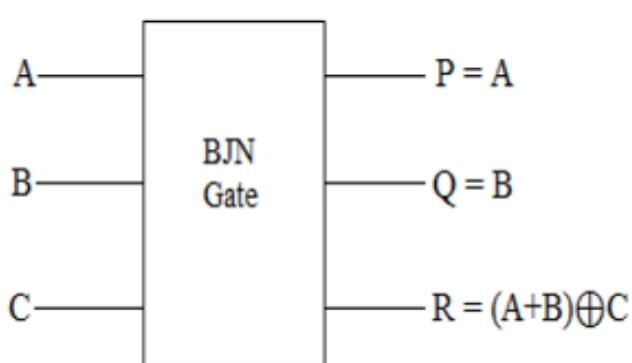


Figure 5.16: Symbol for BJT Gate

A	B	C	P	Q	R
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	1
0	1	1	0	1	0
1	0	0	1	0	1
1	0	1	1	0	0
1	1	0	1	1	1
1	1	1	1	1	0

Table 5.6: Truth Table for BJT Gate

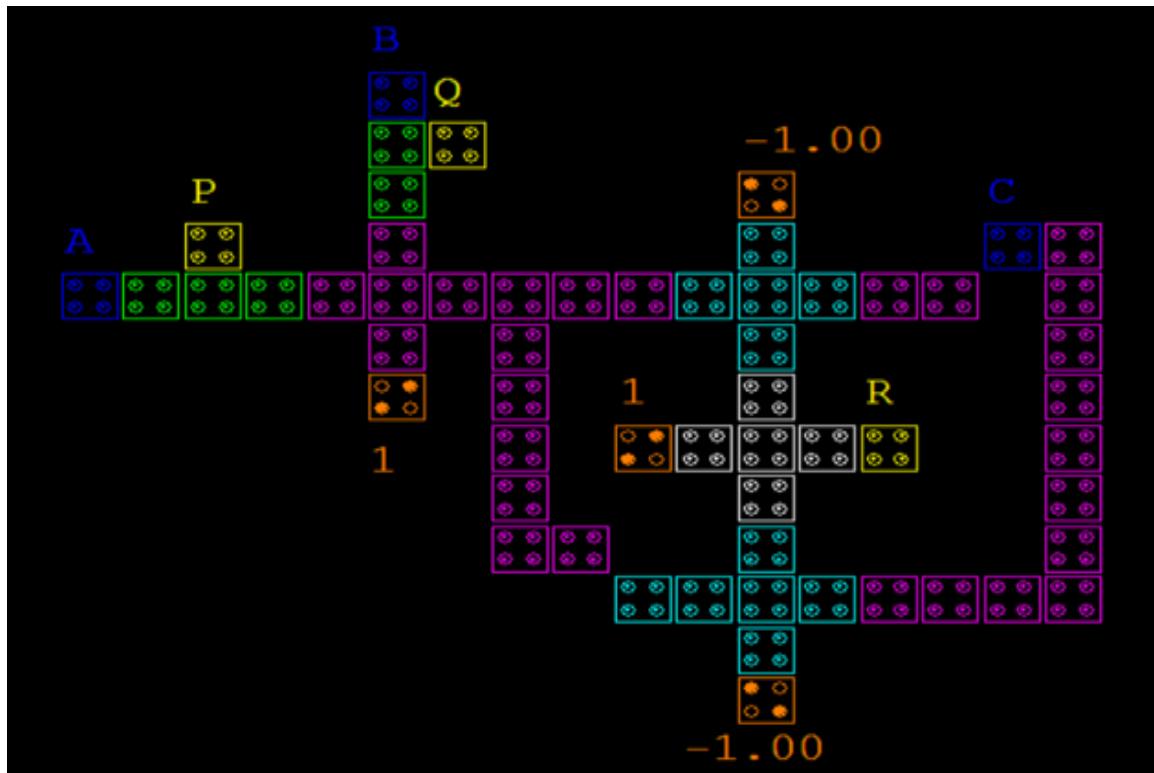


Figure 5.17: QCA Layout of BJT Gate

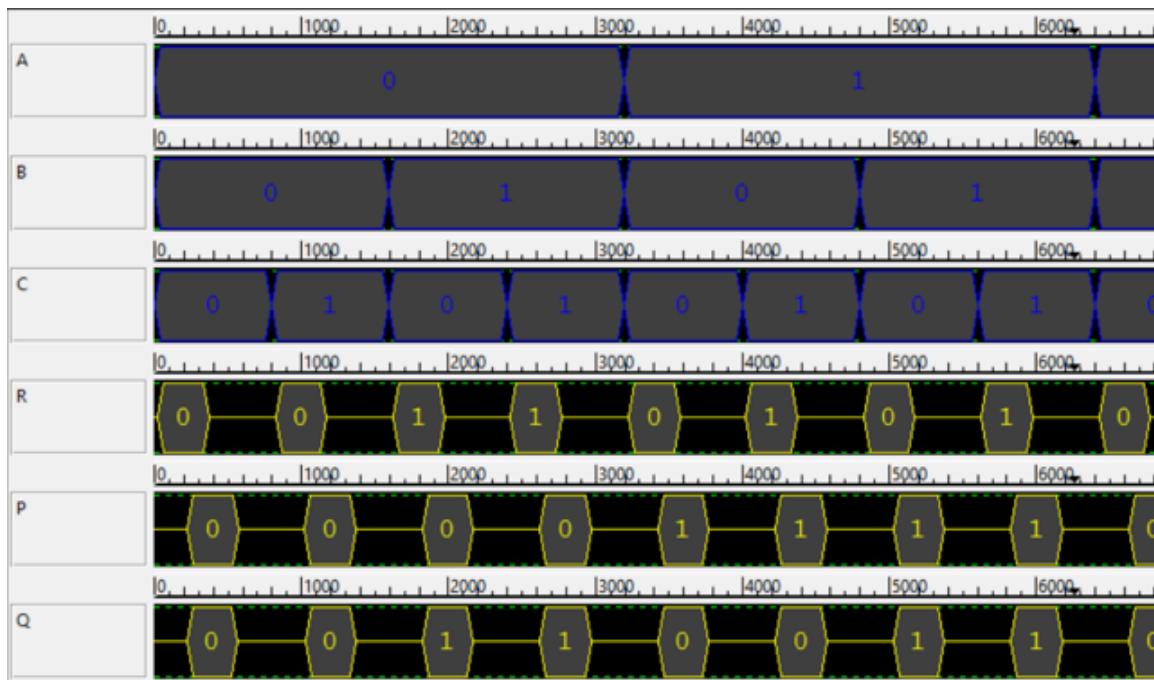


Figure 5.18: Simulation Result of BJT Gate

5.3.7 MCL Gate

This is another 3-input, 3-output reversible gate. At output R, we have input A. Output P is providing logical NOR of inputs B & C while output Q is providing same logical NOR operation for inputs A & B. Basic Symbol and truth table for MCL Gate is shown in Fig. 5.19 and Table 5.7 respectively.

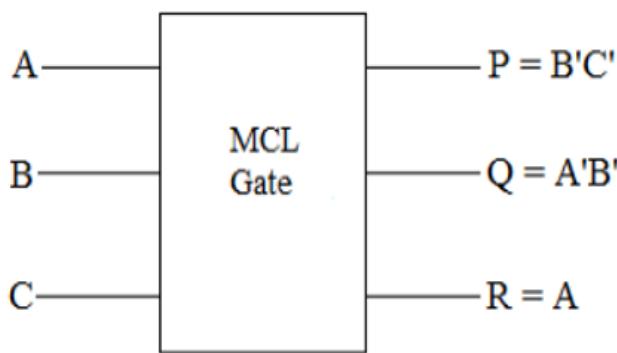


Figure 5.19: Symbol for MCL Gate

A	B	C	P	Q	R
0	0	0	1	1	0
0	0	1	0	1	0
0	1	0	0	0	0
0	1	1	0	0	0
1	0	0	1	0	1
1	0	1	0	0	1
1	1	0	0	0	1
1	1	1	0	0	1

Table 5.7: Truth Table for MCL Gate

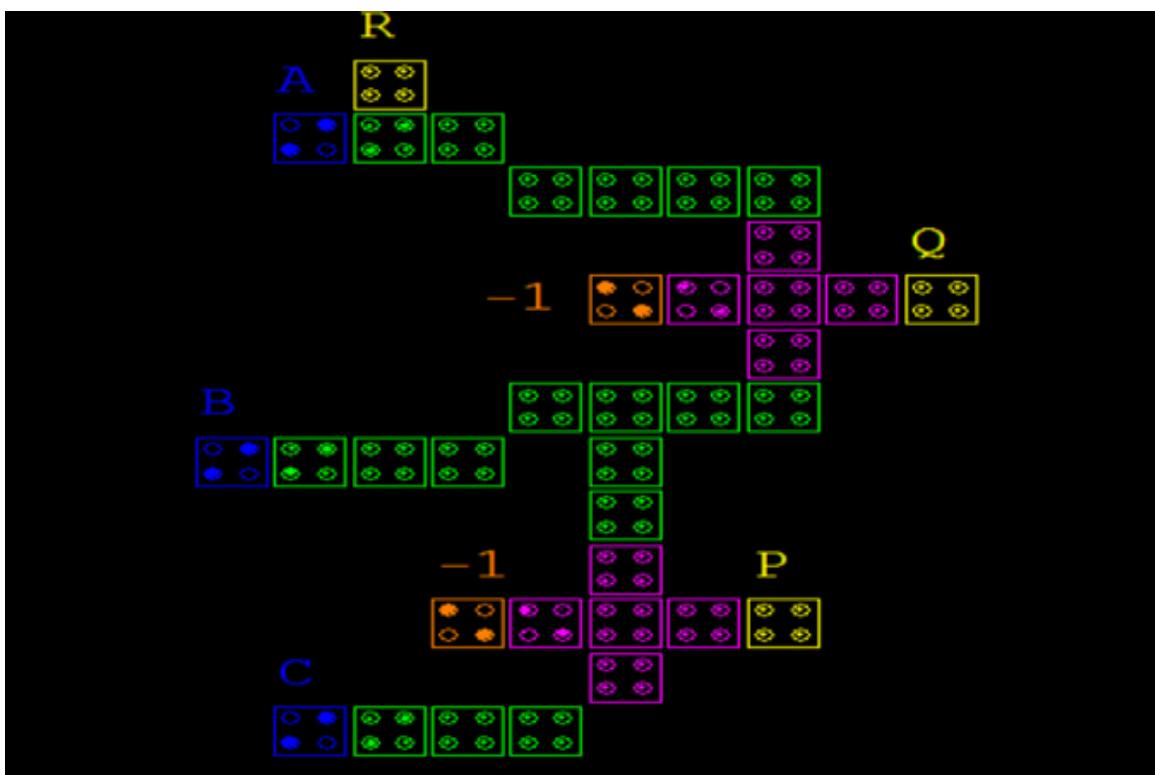


Figure 5.20: QCA Layout of MCL Gate

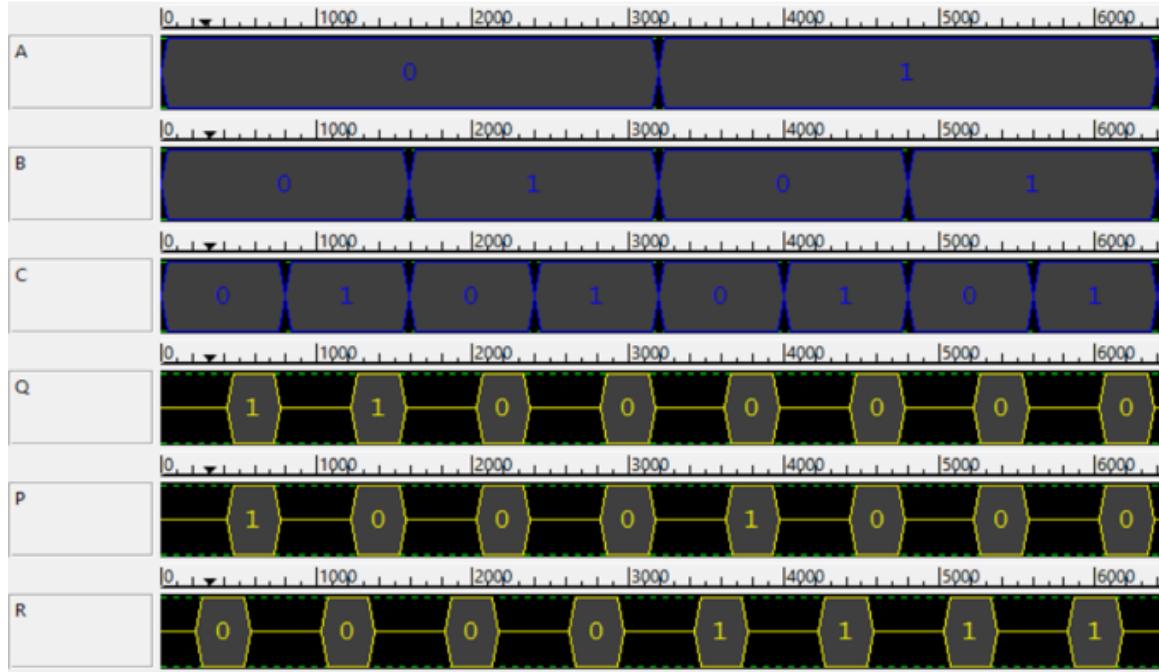


Figure 5.21: Simulation Result of MCL Gate

All these designs were simulated and verified by using QCA Designer tool 2.0.3 under coherence vector simulation engine. The simulation engine parameters are same as Fig. 3.31.

5.4 Performance Analysis of Reversible Gates

Logic Gate	Number of Cells	Area (μm^2)	Latency (Clock Pulse)	Number of Clock	Quantum Costing
CNOT	49	0.06	0.75	3	1
FEYNMAN	53	0.07	0.75	3	1
Double FEYNMAN	93	0.19	0.75	3	2
TOFFOLI	57	0.06	0.75	3	5
FREDKIN	97	0.10	0.75	3	5
BJN	58	0.09	1	4	5
MCL	36	0.05	0.5	2	3

Table 5.8: Performance analysis of Reversible Gates

Chapter 6

Proposed Designs and Their Implementation

6.1 Adder Designing

As we have seen some of the adder design implemented and simulated successfully in chapter 3. Here we are going to propose 1-Bit adder and 2-Bit adder designs.

6.1.1 1-Bit Adder

A 1-Bit Adder circuit computed addition of two numbers along with the carry input. Carry is taken from the flag register and after the computation, depending upon the result; flag register is either set or reset.

In this design we use two-wires, one is normal and another at the surface, which cross over together, with a new type of crossover known as logical crossover. In this crossover, two wires are clocked in such a manner that they are 180^0 out of phase so that they pass each other at different time slot and no interference occurs. Over the intersection point Central cell will follow only wire of clock phase and in result there is only two combination of clock phases at the at the intersection point which confront.

During the simulation, when first wire is in switch phase the other wire remains in release phase. Therefore, the former one is getting the polarization from the previous cell while the latter one is losing its polarization. By this, it can be concluded that only original wire polarization is important, and second wire's state doesn't affect this. On the other hand, when these wires maintain another polarization possibility i.e. Hold and Relax. The cell moment of first wire is fixed and passing it to the next cell. While the second wire cells are relaxed and having NO polarization, so will have no impact on performance.

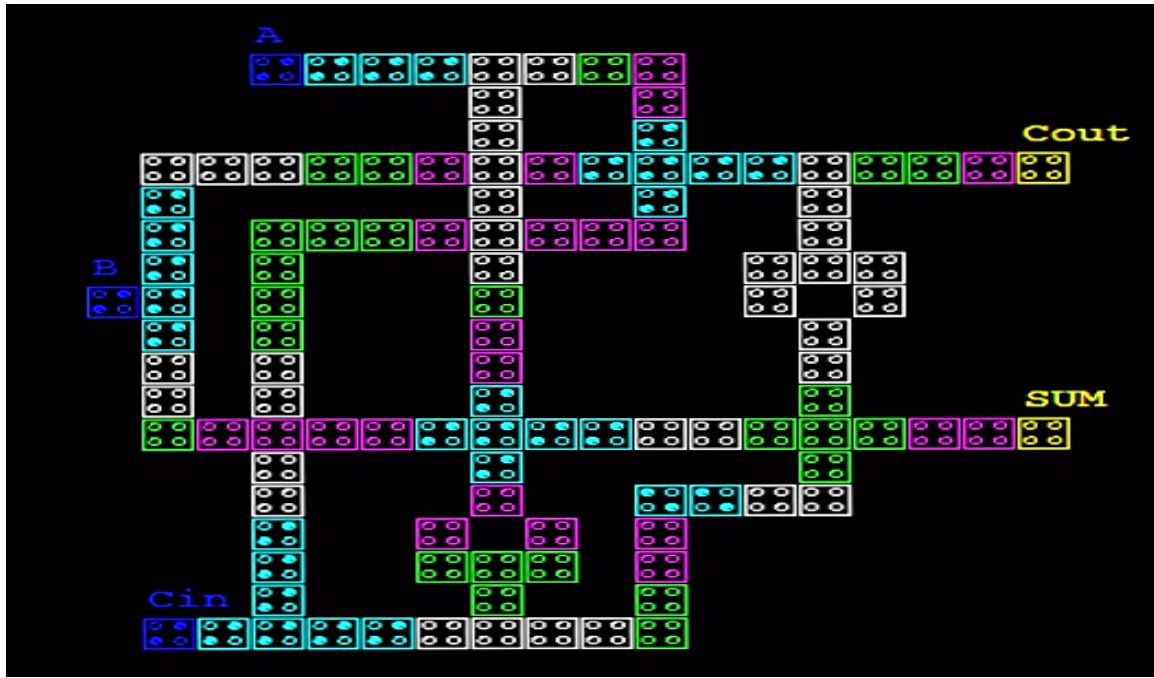


Figure 6.1: QCA Layout of 1-Bit Adder

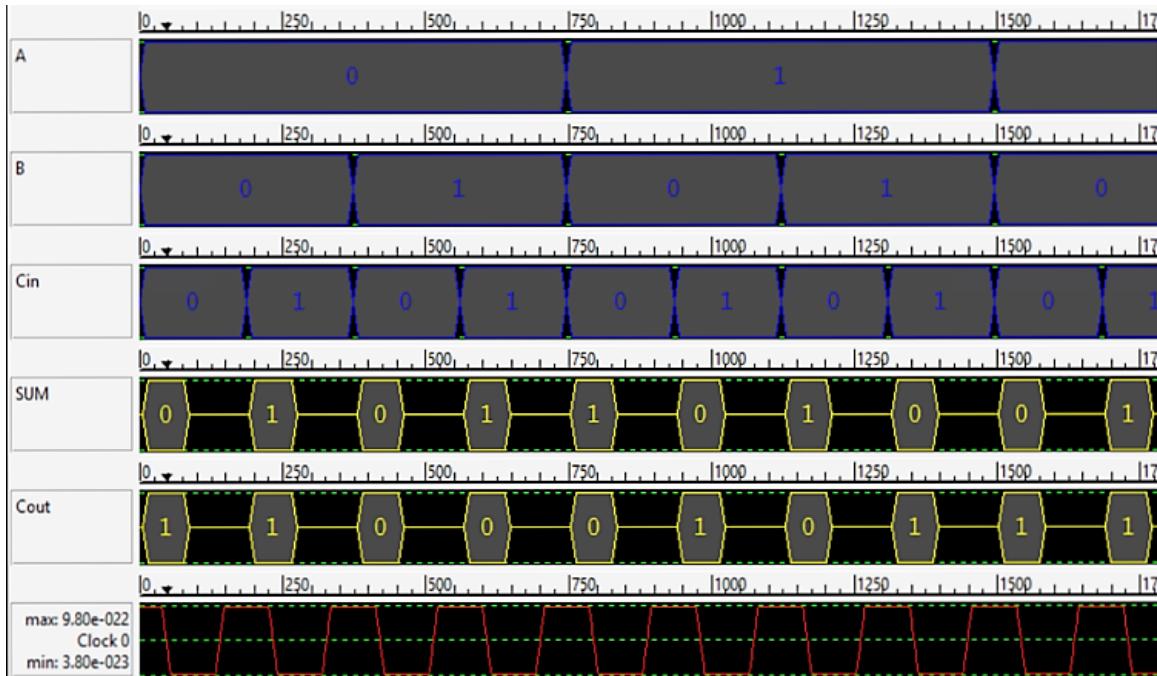


Figure 6.2: Simulation Result of 1-Bit Adder

Although this design is working fine but to reduce the number of cells and as well area, we can replace certain areas as shown below

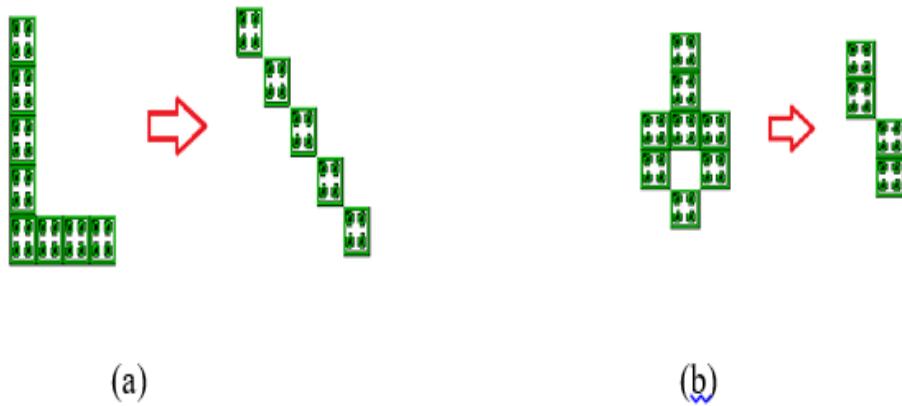


Figure 6.3: Cell reduction techniques (a)Diameter Wire (b)Stair Inverter

6.1.2 1-Bit Adder Modified Design

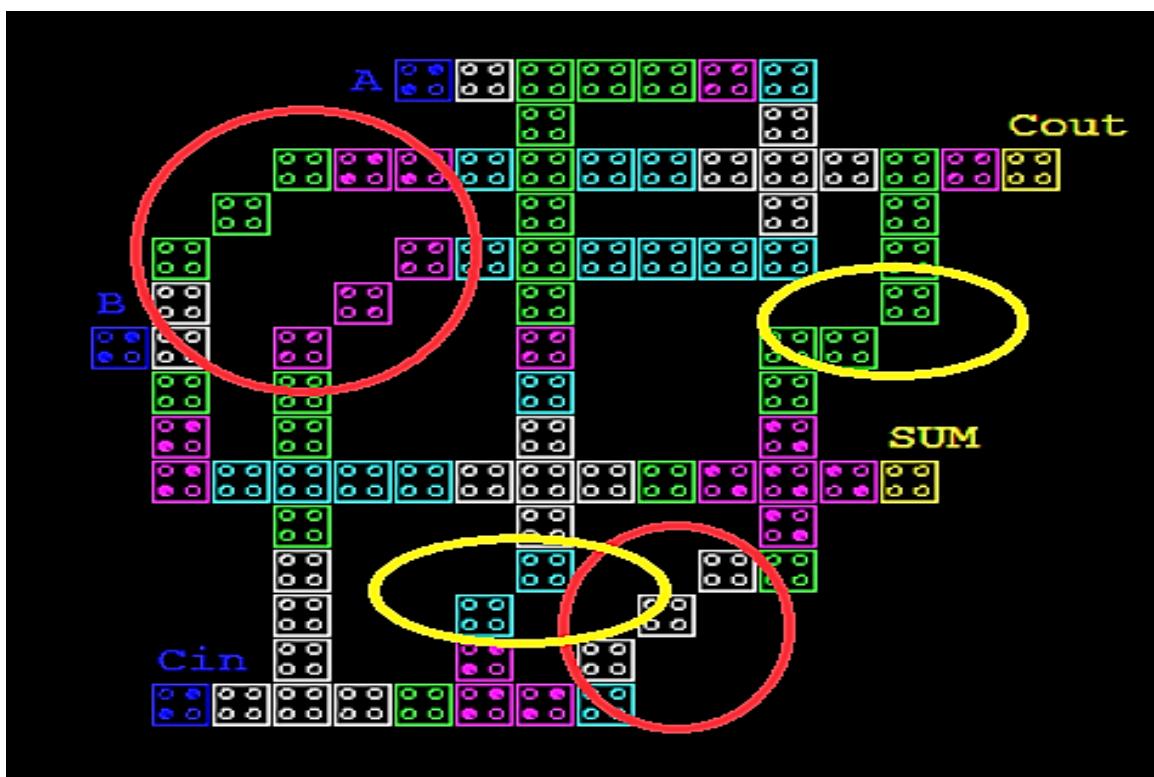


Figure 6.4: QCA Layout of Modified 1-Bit Adder

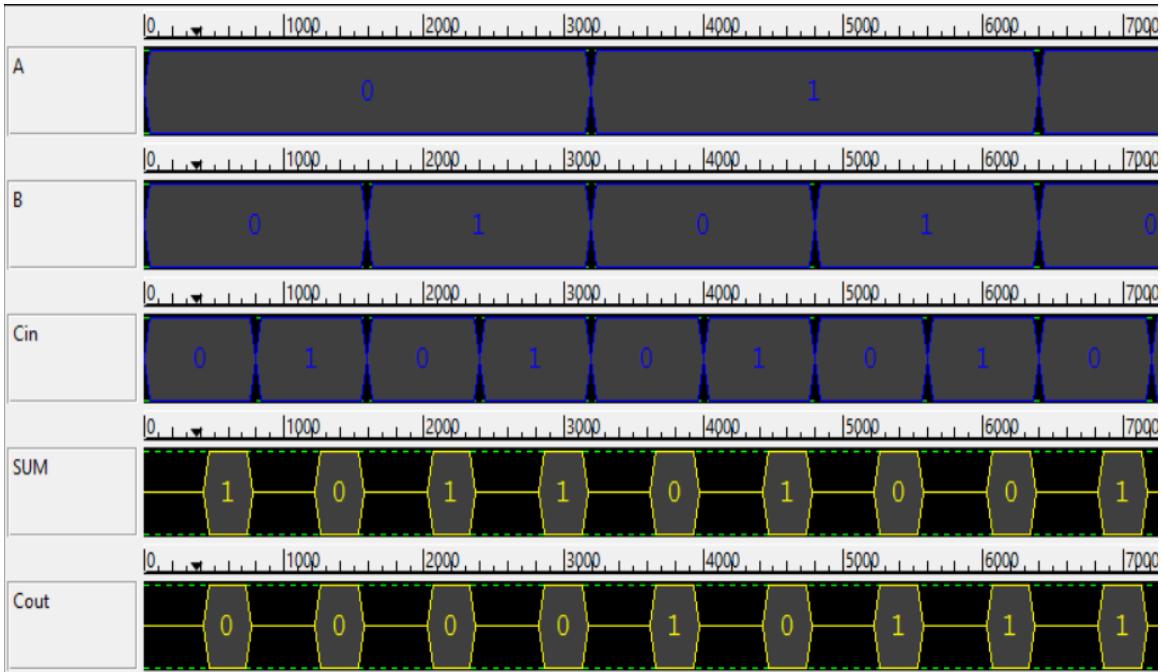


Figure 6.5: Simulation Result of Modified 1-Bit Adder

As shown in Fig. 6.4, Yellow coloured boxes show replacement of Robust inverter with stair inverter and Red coloured boxes show replacement of corner wires with diameter wires as proposed in Fig. 6.1.

6.1.3 Comparison Study

Design	Number of Cells	Area (μm^2)	Latency (Clock Cycles)
Basic Design	115	0.15	2
Modified Design	87	0.10	1

Table 6.1: Comparison of Adder Designs

As shown in Table 6.1, modified design has a significant improvement in area and cell count. The Number of cells and area are reduced by 32% and 33% respectively in modified design with a faster response of 1 clock cycle.

6.1.4 2-Bit Adder

Another advantage of modified design is that we can extend it to higher order designs. In this section, we have used two 1-bit adders to design a 2-bit adder. They are used in a manner that C_{out} of the first stage is C_{in} of second stage. Second stage inputs are provided after the computation of first stage gets

over so that result proper can be achieved. The Same strategy is applied while collecting outputs; S_0 is delayed for period required for the second stage to complete its operation.

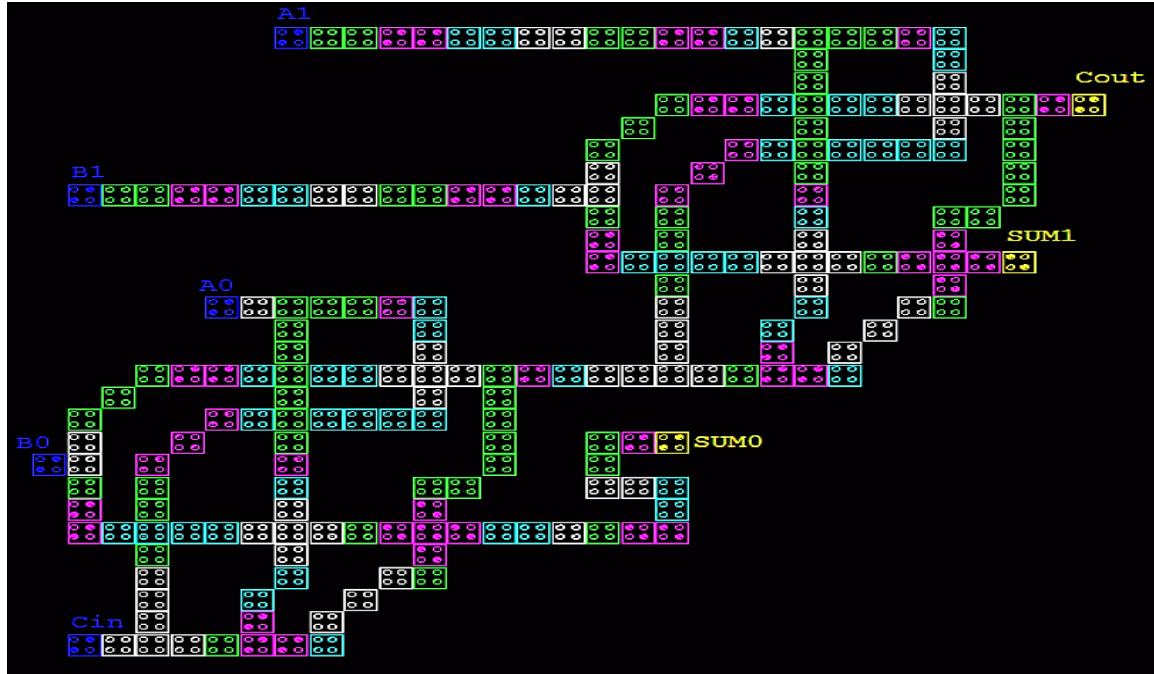


Figure 6.6: QCA Layout of 2-Bit Adder

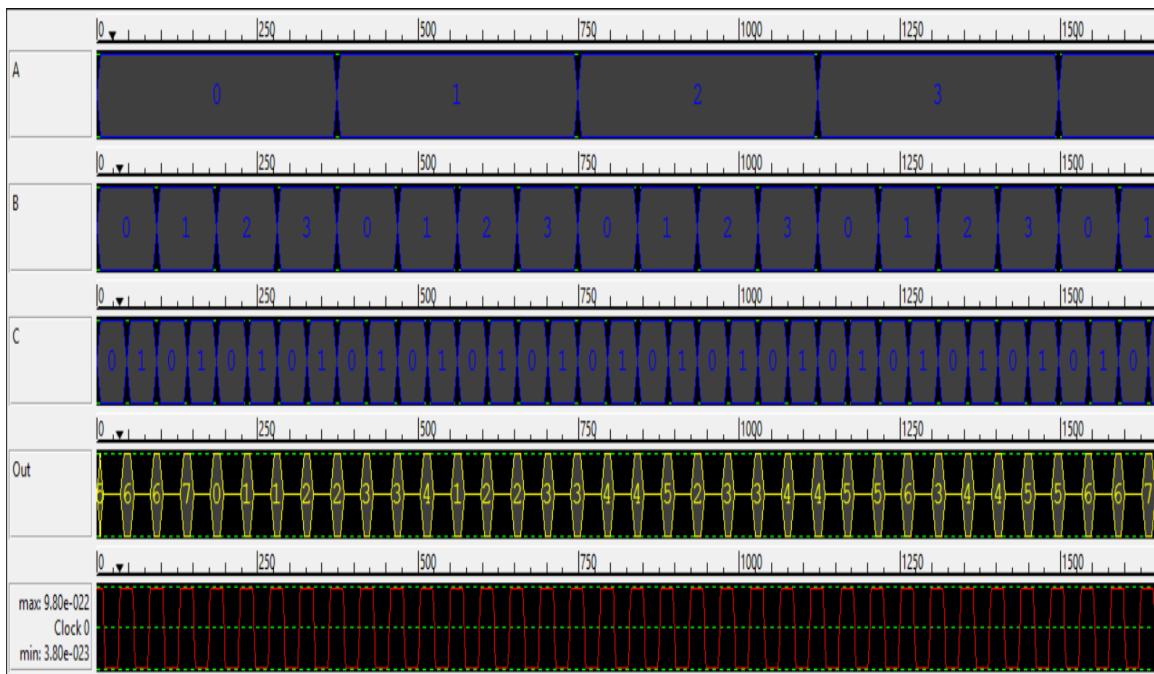


Figure 6.7: Simulation Result of 2-Bit Adder

This design is simulated and verified by QCA Designer tool 2.0.3 under coherence vector simulation engine. Parameters for coherence vector engine are as same as Fig. 3.31. As shown in Fig. 6.7, 2-Bit Adder circuit has 3.25 clock cycles delays and uses 218 cells in an area of $0.38 \mu\text{m}^2$.

6.1.5 Power Analysis of Adder Designs

As we have proposed designs for 1-bit and 2-bit adder, in this section we have tried to find out power dissipation analysis for those designs.

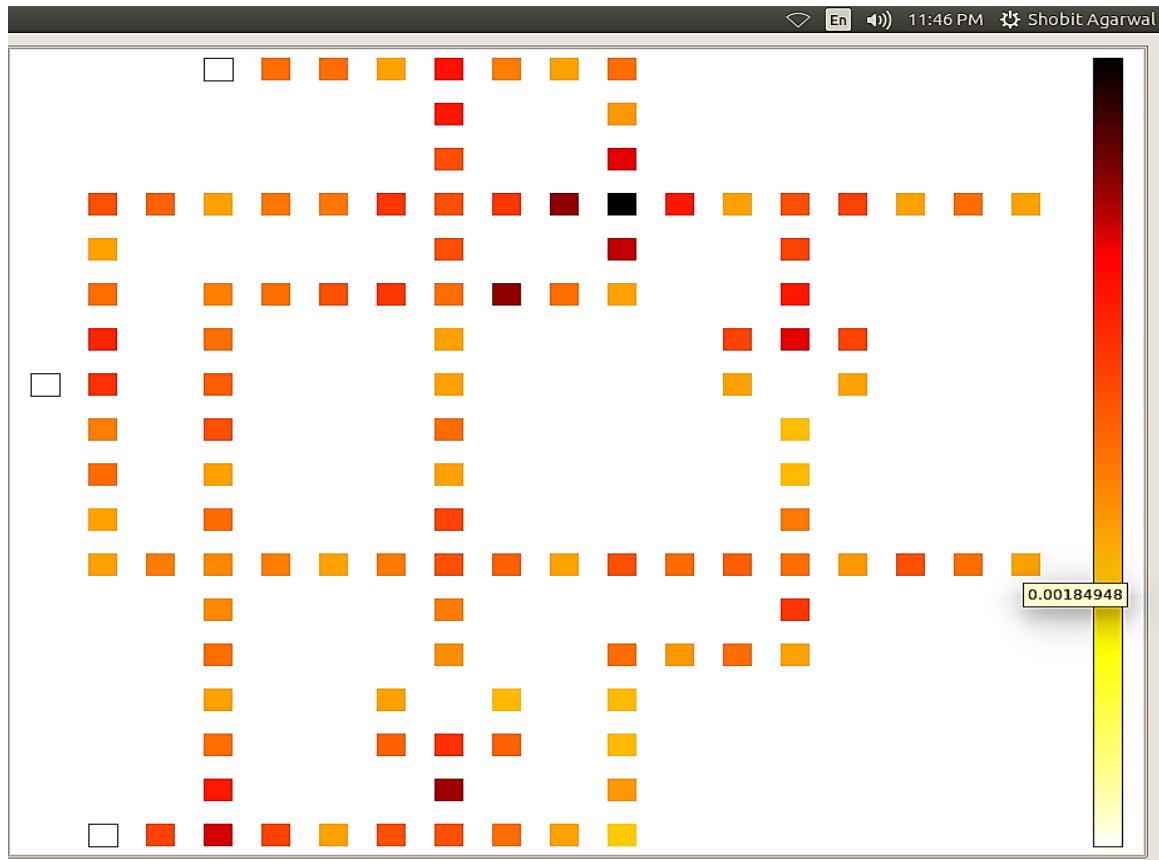


Figure 6.8: Power dissipation map for 1-bit adder ($1, 0.5 E_k$)

Comparison analysis for these designs along with other designs is shown in Chapter 7.



Figure 6.9: Power dissipation map for 1-bit adder modified (1, 0.5 E_k)

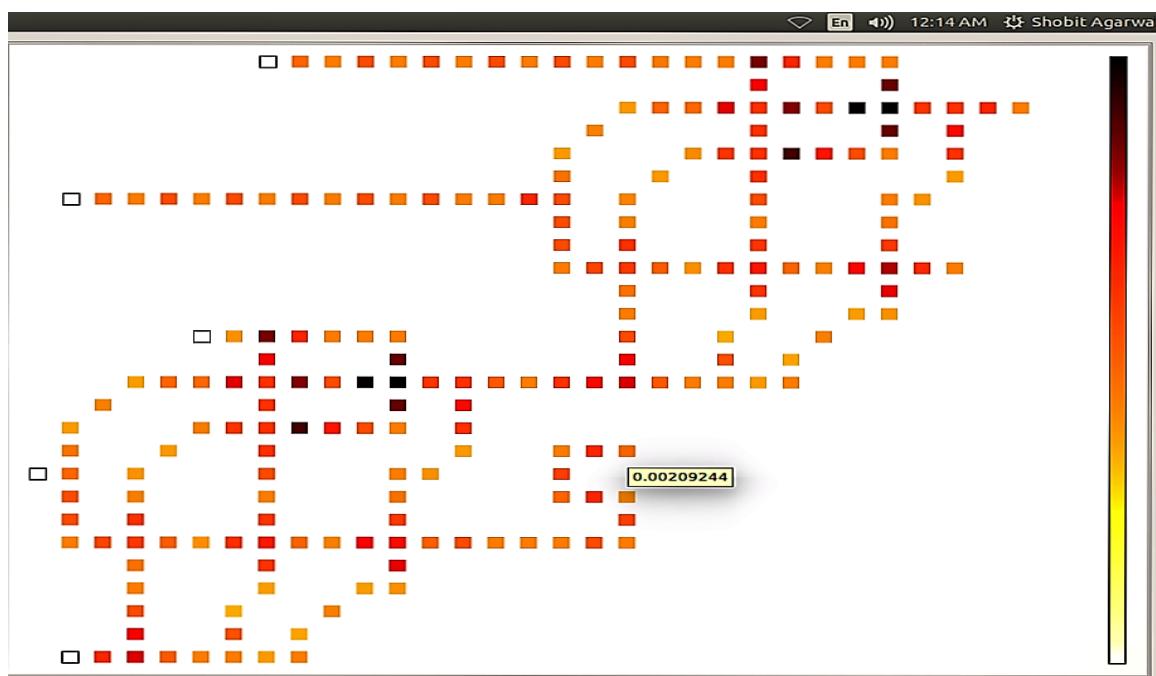


Figure 6.10: Power dissipation map for 2-bit adder (1, 0.5 E_k)

6.2 Adder/Subtractor Designing

Adder/ Subtractor circuit is a combination of adder and subtractor, which means it can perform addition as well as subtraction. In order to select operation to be performed, we need to have a *control* signal. For each state of *control* signal, one operation can be performed (Let's say for addition it is "0" while for subtraction it is "1"). As proposed in [23], author has proposed a controllable inverter made up of XOR gates. In combination to this stage, 8 Full adders are also made and they form 1st stage of circuit. 2nd stage is also the replica of first stage with control input as C_{out} of first stage.

In proposed design we have used the same algorithm but with new XOR gate based controllable inverter. New XOR gate comprises only 30 cells and uses an area of $0.02 \mu m^2$. QCA Layout and Simulation results of this XOR Gate is as shown below:

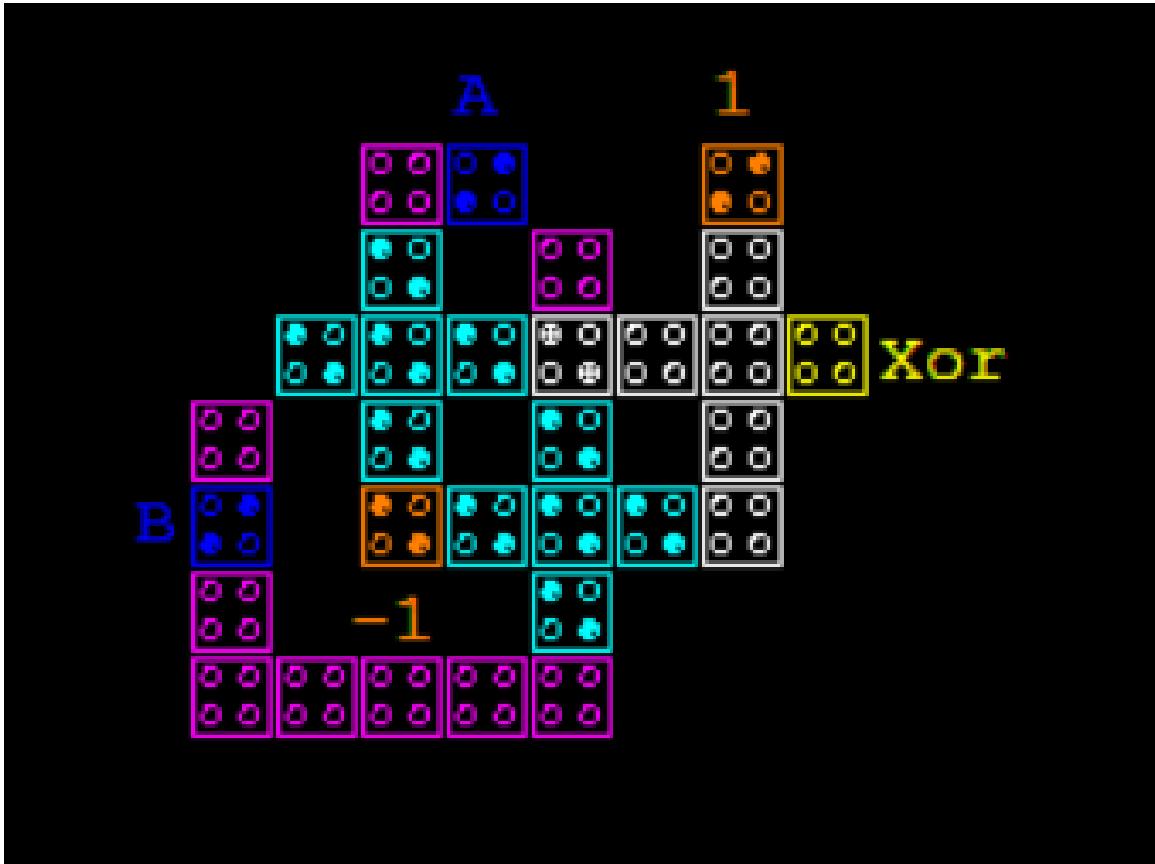


Figure 6.11: QCA Layout of XOR Gate

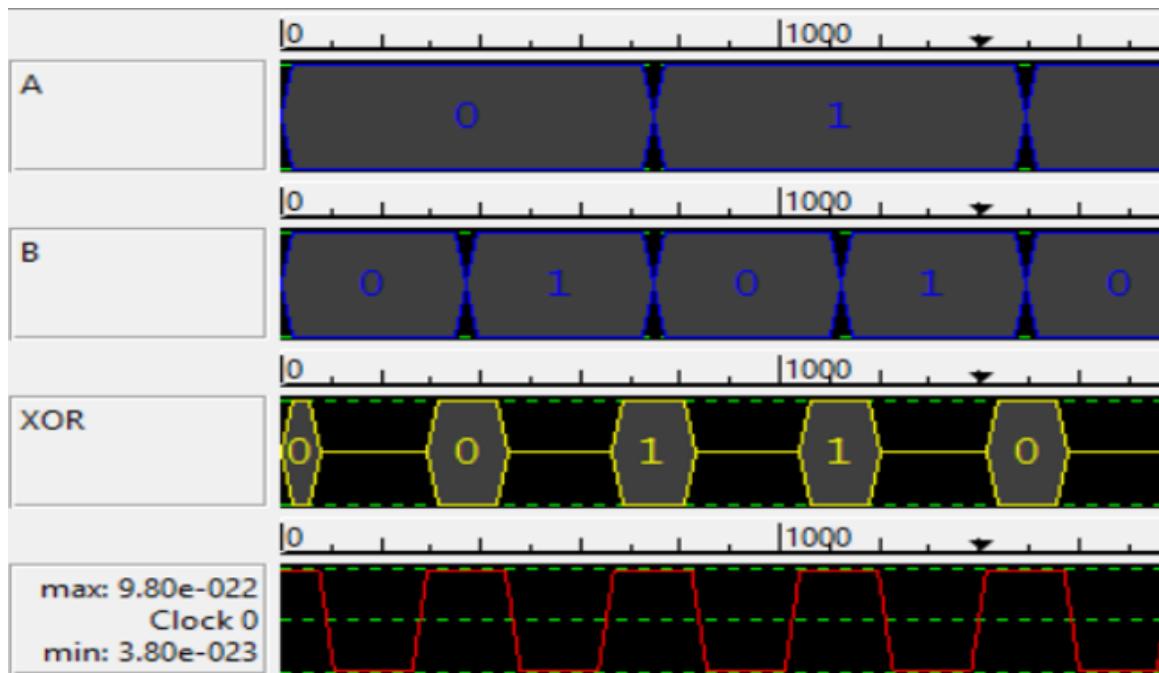


Figure 6.12: Simulation Result of XOR Gate

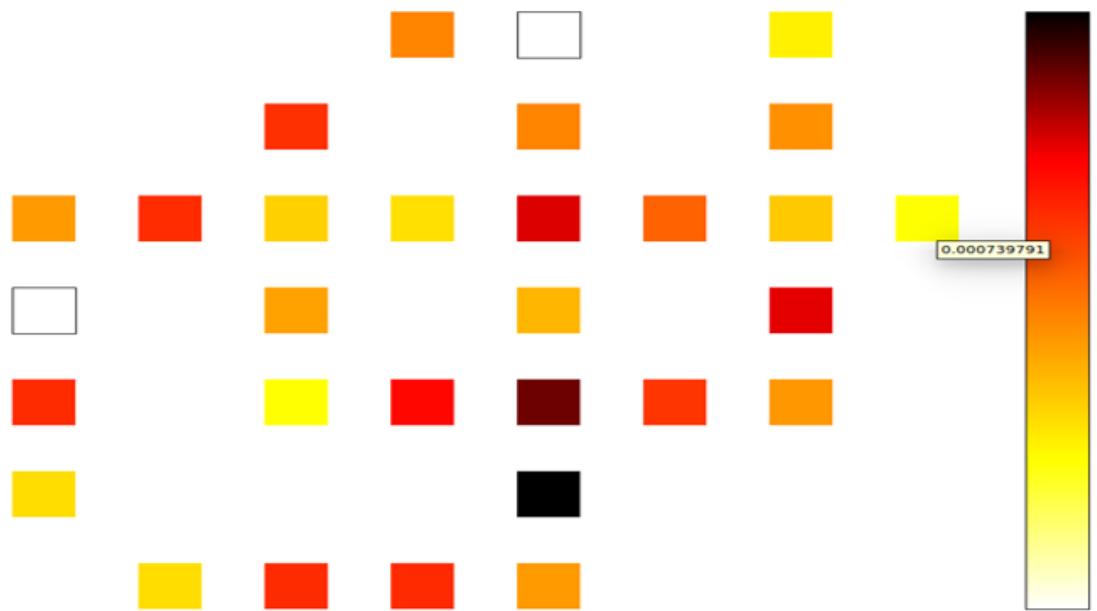


Figure 6.13: Power dissipation map for XOR Gate ($1, 0.5 E_k$)

Kink Energy	Tempearture	
	1	2
(E5)	0.000739791	0.000739514
1	0.00147958	0.00147855
1.5	0.00221937	0.00221852

Table 6.2: Power dissipation values for XOR Gate

Based upon this XOR gate, an eight bit controllable inverter is made. Concept of XOR gate says that it works as an inverter if one of its two inputs is tied to logic “1” and it works as a buffer if one of its two inputs is tied to logic “0”. Hence if we tie one input of all XOR gates together and name it as *control* input then we can use either form of second input. If *control* input is zero, then output is as same as input while if *control* input is “1”, output is inverted form of input.

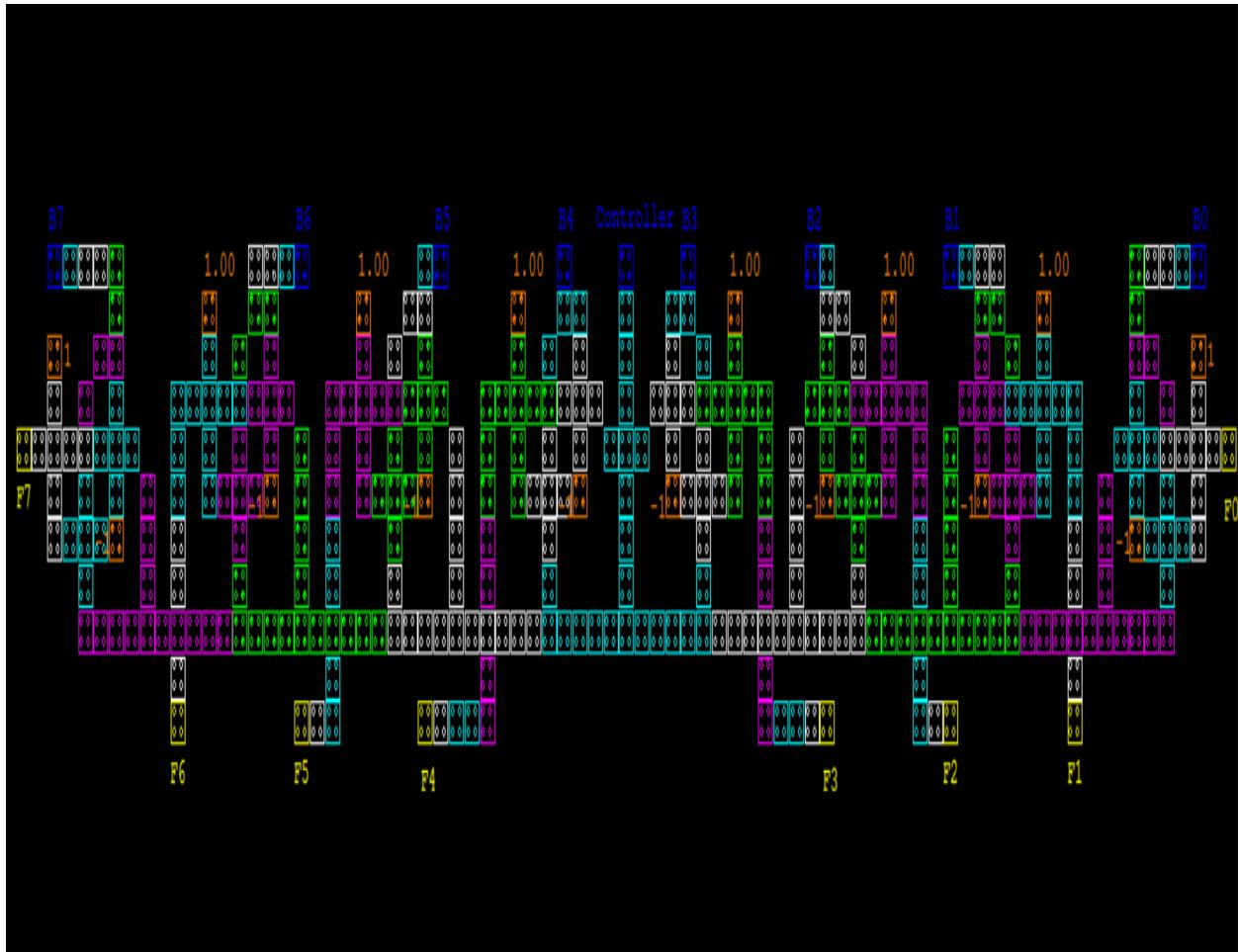


Figure 6.14: QCA Layout of 8-bit Controllable inverter

Inputs	Active	0	1	2	3	4	5	6	7	8	9	10	11	12
Controller	<input checked="" type="checkbox"/>	0	0	0	1	1	1	0	0	0	0	0	1	1
Controller	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
B[7:0]	<input checked="" type="checkbox"/>	83	52	229	1	88	162	37	72	50	173	82	44	0
B7	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
B6	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
B5	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
B4	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
B3	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
B2	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
B1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
B0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Figure 6.15: Simulation Vector of 8-bit Controllable inverter

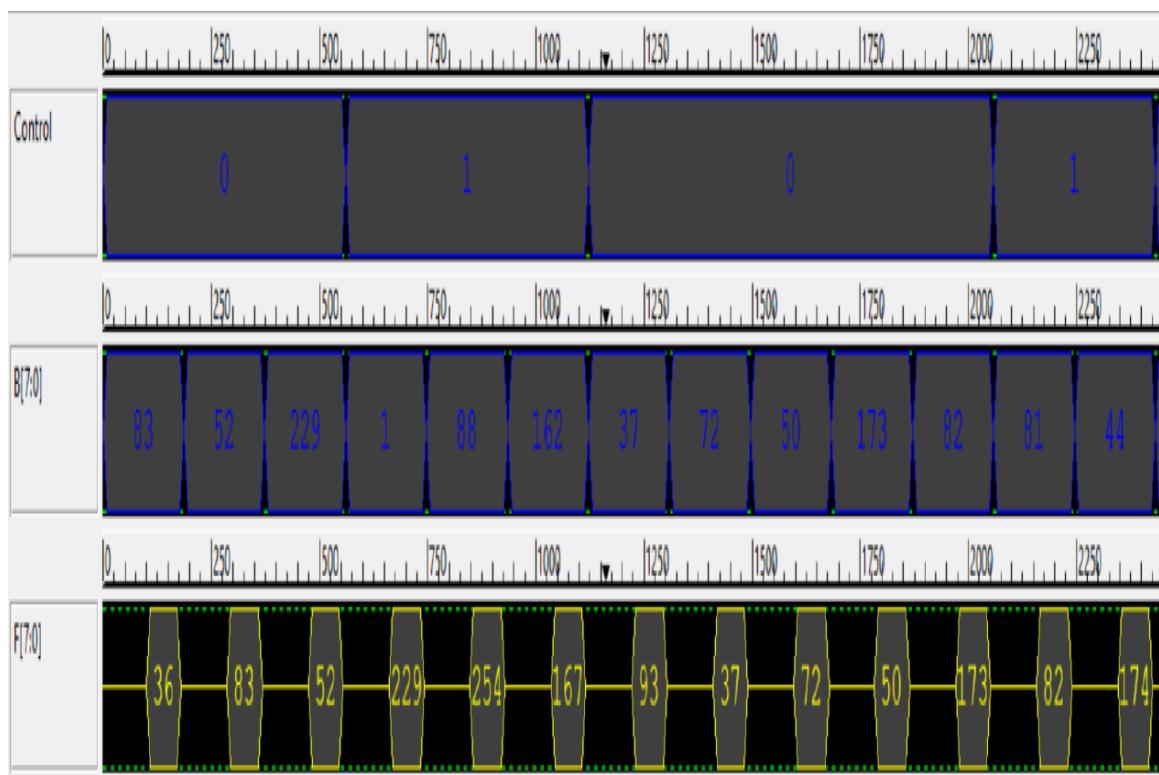


Figure 6.16: Simulation Result of 8-bit Controllable inverter

Now, in order to get an adder/ subtractor circuit, we have followed procedure as shown in Fig. 6.16. In order to add two 8-bit numbers (say A and B here), *control* input is set to zero so that input B (applied to one input of XOR gates) is transmitted directly towards output. From here it is fed to one bit full adder circuit chain where other inputs are A and C_{in} (obtained from control input). Hence addition of A & B is fed to second stage. Since *control* input is zero, C_{in} for next stage is again zero and A+B is directly forwarded to output.

For subtraction, the subtrahend input should be applied to controllable inverter and minuend is applied to full adder inputs. Now, let B is subtrahend and A is minuend. To subtract B from A, firstly we need to obtain 2's complement of B. it is obtained by setting Control input to “1”. By this controllable inverter provides 1's complement of B. Now in full adders, A+B+1 is computed (as $C_{in}=1$ due to Control input) which is actually A-B, and it is propagated to second controllable inverter. If C_{out} of 1st stage is “1”, it shows that A>B and result is valid. Hence inverse of this C_{out} is logically AND with *control* input and given to second stage. Since output of AND gate is zero so the input is directly forwarded to output.

On the other hand, if C_{out} of first stage is “0”, it shows A<B and we need to compute 2's complement of result. Again inverse of C_{out} is logically ANDed with *control* input and as both inputs are “1” so for second stage again *control* input is “1” and performs a 2's complement of 1st stage result.

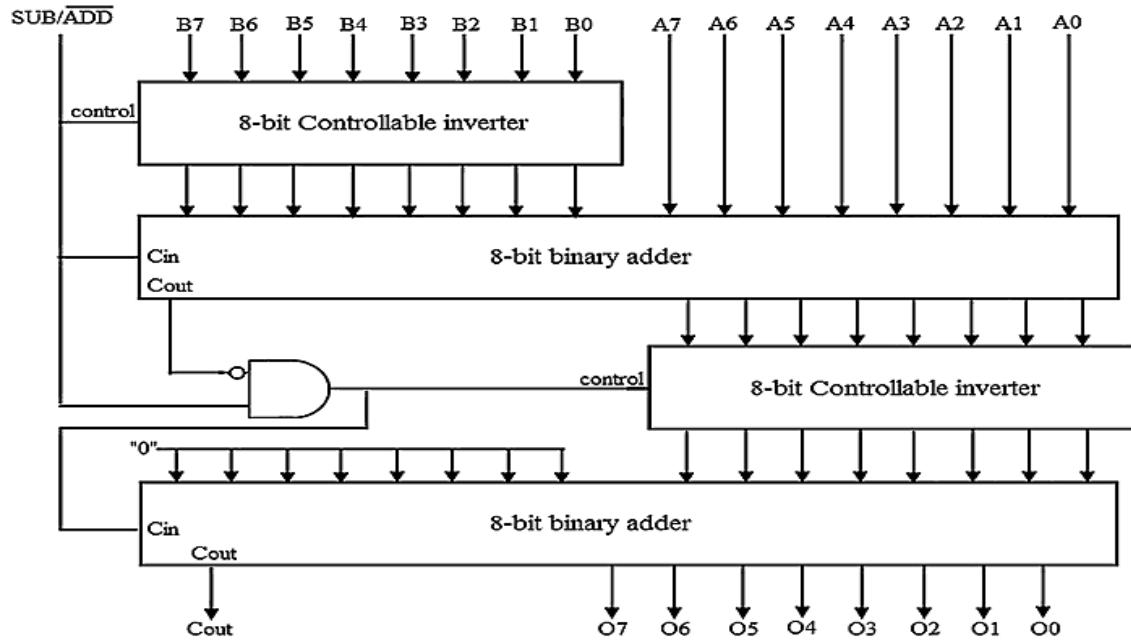


Figure 6.17: Adder/Subtractor Algorithm proposed in [23]

QCA Layout and simulation result for 8-bit adder/ subtractor circuit is shown in Fig 18 and Fig. 20 respectively. This layout uses a total of 2429 QCA cells in area of $2.46 \mu\text{m}^2$. As clear from simulation result that this circuit is having a propagation delay (Latency) of 8.75 clock pulses.

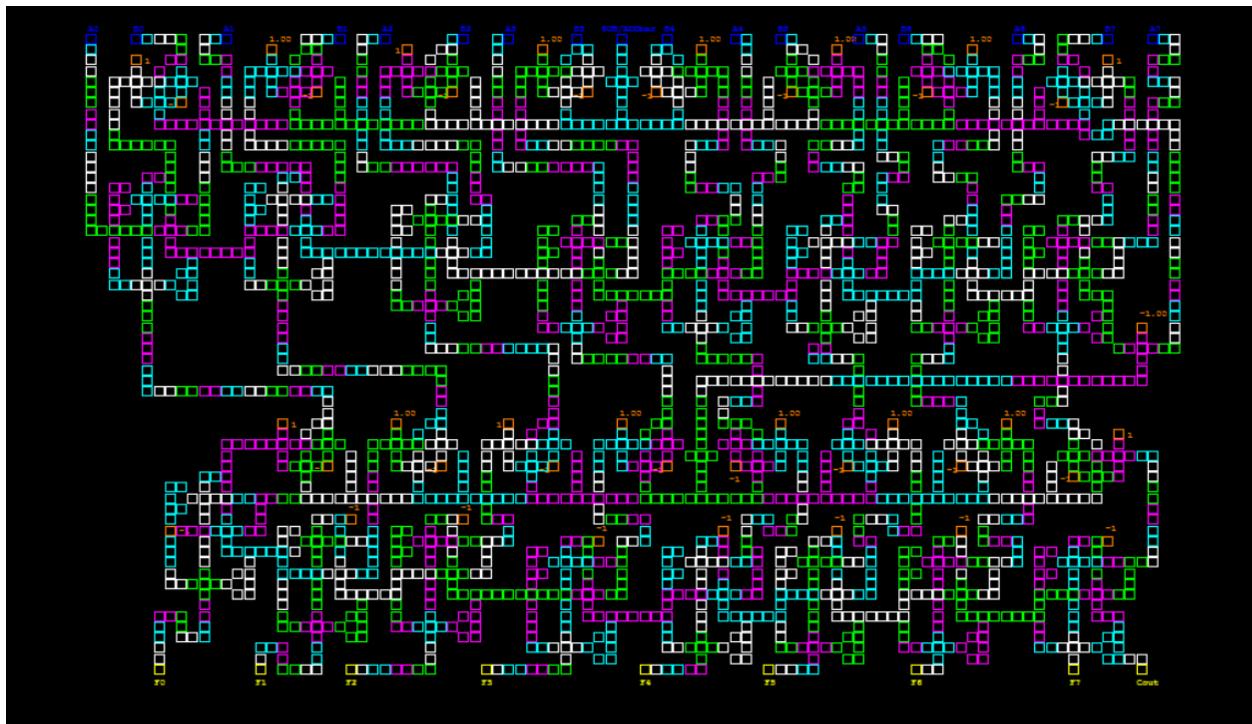


Figure 6.18: QCA Layout of 8-bit Adder/Subtractor

Inputs	Active	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
☛ SUB/ADDbar	☒	0	0	0	1	1	0	1	1	0	0	0	0	0	0	0
☛ SUB/ADDbar	☒	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
☛ A[7:0]	☒	139	246	64	255	109	64	154	205	0	0	0	0	0	0	0
☛ A7	☒	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
☛ A6	☒	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
☛ A5	☒	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
☛ A4	☒	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
☛ A3	☒	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
☛ A2	☒	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
☛ A1	☒	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
☛ A0	☒	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
☛ B[7:0]	☒	26	9	125	55	99	36	54	200	0	0	0	0	0	0	0
☛ B7	☒	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
☛ B6	☒	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
☛ B5	☒	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
☛ B4	☒	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
☛ B3	☒	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
☛ B2	☒	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
☛ B1	☒	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
☛ B0	☒	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 6.19: Simulation Vector of 8-bit Adder/Subtractor

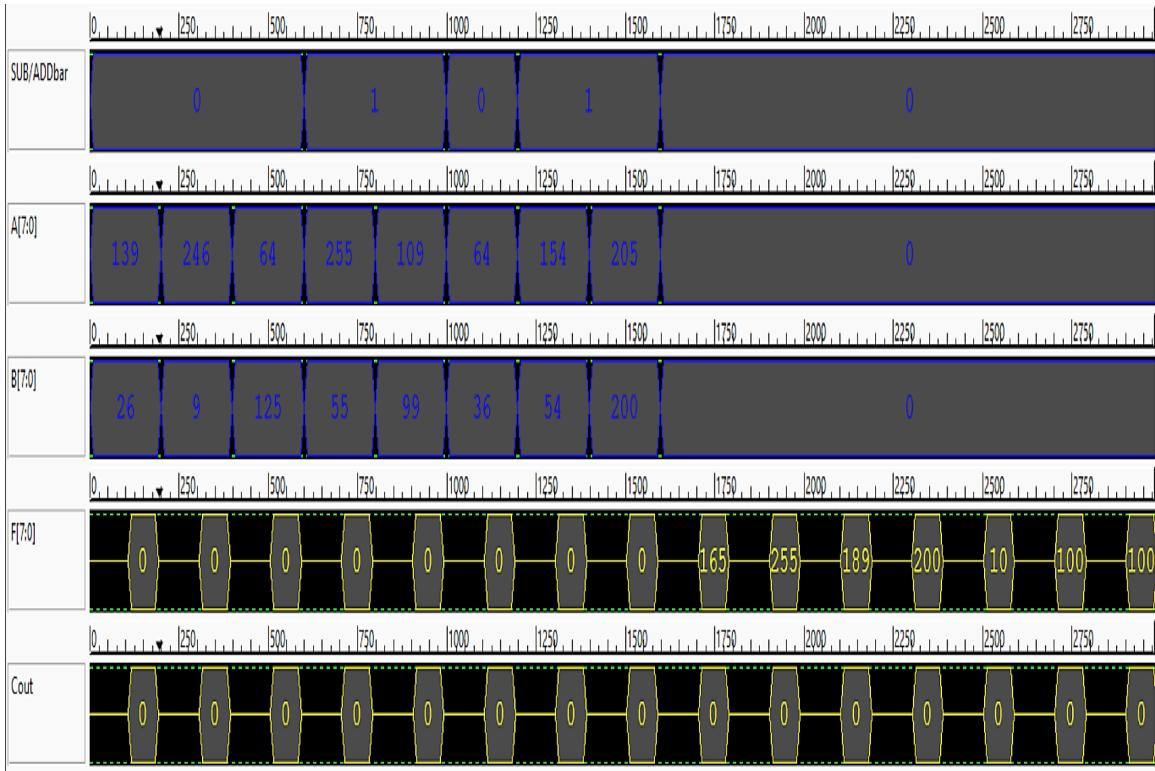


Figure 6.20: Simulation Result of 8-bit Adder/Subtractor

Design	Number of Cells	Area (μm^2)	Latency (Clock Pulse)
In [23]	5786	10.3	27.25
Proposed	2429	2.46	8.75

Table 6.3: Comparative Study of adder/subtractor designs

In Comparison to design proposed in [23], this design has improvement of 318% in area, 138% in cell count and 210% in latency. Hence this is very good design over the design proposed in [23].

Chapter 7

Power Analysis in QCA

7.1 Power dissipation in QCA

For power calculations in QCA a tool known as QCA Pro is suggested which takes on a non-adiabatic power dissipation model to estimate switching power losses in a QCA based circuit. And this model comes from Timler's quasi-adiabatic model, by this equation for instantaneous power is given as[30]:

$$P_{Total} = \frac{dE}{dt} = \frac{\hbar}{2} \left(\frac{d}{dt} \vec{\Gamma} \right) \cdot \vec{\lambda} + \frac{\hbar}{2} \cdot \vec{\Gamma} \left(\frac{d}{dt} \vec{\lambda} \right)$$

Where λ = Coherence Vector; Γ = Three dimensional energy vector.

The First term defines the power in and out of the clock and cell to cell power flow. The Second term represents instantaneous power dissipation. This occurs during switching so can be calculated by integrating P_{diss} over time.

$$P_{diss}(t) = \frac{\hbar}{2} \cdot \vec{\Gamma}(t) \cdot \left(\frac{d}{dt} \vec{\lambda} \right)$$

7.1.1 Modelling Scheme

There is a causal input-output behavior in clocked QCA architecture, so a graphical probabilistic framework is best to represent underlying dependencies. QCA Pro comes here in picture; it converts a QCA circuit into a Bayesian network. The node of Bayesian networks represents individual QCA cell and edges represents the flow of information from input nodes to output nodes. The conditional probability of individual nodes is given by steady state polarization probability equation as[30]:

$$P^{ss} = -\lambda_3^{ss} = \rho_{11}^{ss} - \rho_{00}^{ss} = \frac{E_k \cdot \bar{P}}{\sqrt{(E_K^2 \cdot \bar{P}^2 + 4\gamma^2)}} \tanh(\sqrt{\frac{E_K^2 \cdot \frac{\bar{P}^2}{4+\gamma^2}}{KT}})$$

The Equation can be re-written as:

$$P^{ss} = \frac{E}{\Omega} \tanh(\Delta)$$

Where $E = 0.5 \sum i E_k P_i f_i$, The total kink energy

$\Omega = \sqrt{(\frac{E_K^2 \cdot \bar{P}^2}{4+\gamma^2})}$, The Rabi frequency

$\frac{\Omega}{KT}$ = Thermal Ratio

This probability depends on tunneling energy, the polarization of neighboring cells, temperature, size and spacing of QCA cells.

QCA Pro uses this approach to study the most probable circuit output states, to model the error-proneness of the QCA cells based on polarization probability of a given cell. Fig. Shows a design of a single bit QCA adder circuit and its equivalent Bayesian network[3].

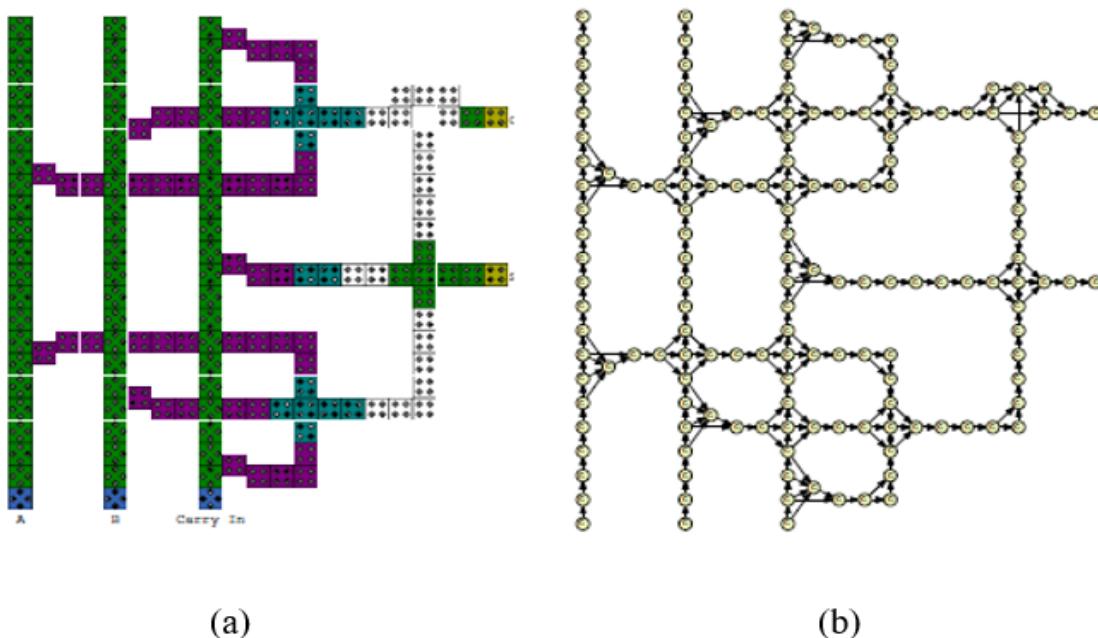


Figure 7.1: 1-Bit Adder design in QCA (a)Cell Layout; (b)Equivalent Bayesian Model

7.1.2 QCA Pro Simulation Setup

QCA Pro can generate a polarization map of QCA circuit for any particular input vector. This step can be repeated several times to generate a map for different input vectors. In order To generate power dissipation map for QCA circuit the user has first to provide the set of input vectors to be switched and the ratio of kink energy to tunnelling energy. QCA Pro then calculates average power dissipation over all specified input vectors and also generates a power dissipation map that displays the thermal hotspots in the design. The tool allows a user to scroll over a particular cell in a QCA circuit to view its polarization probability or power dissipation values. The user can also save the polarization probability and power dissipation map. Fig. 7.2 Shows how this flow takes place.[5]

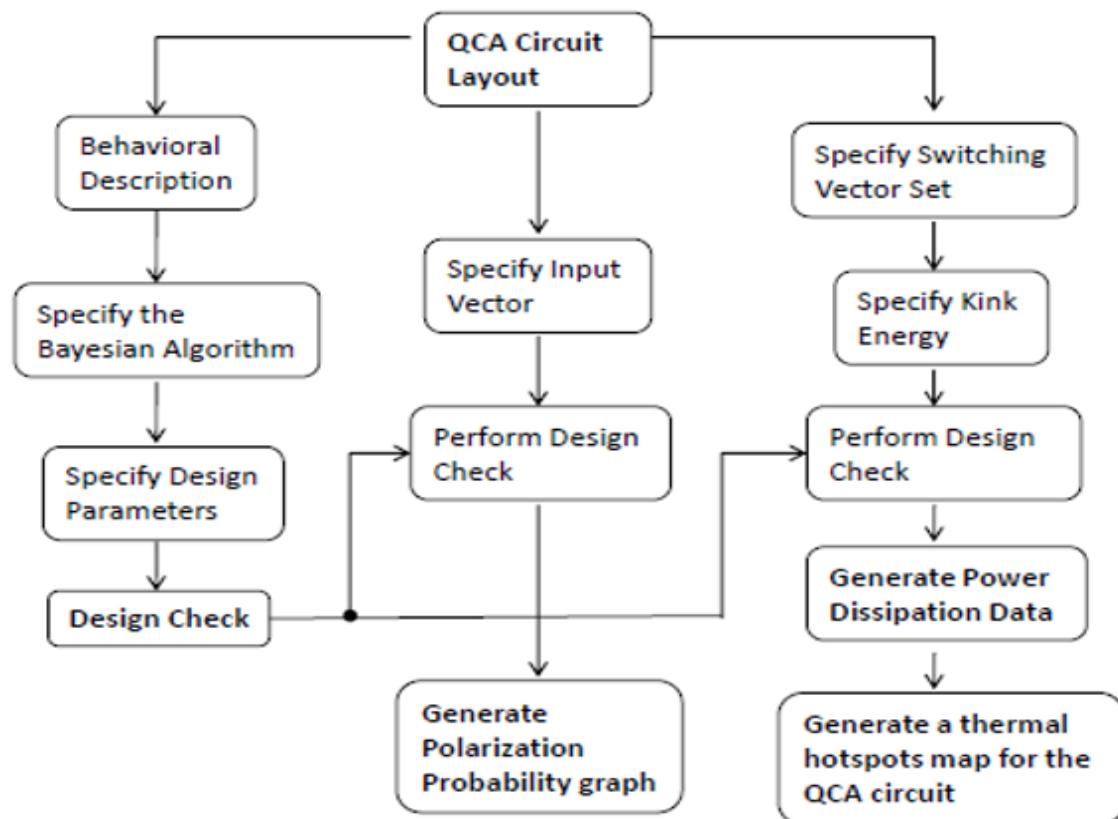


Figure 7.2: Design flow for QCA Pro tool

7.2 Power Analysis of implemented circuits

QCA Pro tool is compatible only with the QCA Designer tool 1.4.0, so all the circuits that implemented using QCA Designer 2.0.3 and shown in previous chapters are again designed for the power calculations within QCA Designer tool 1.4.0.

7.2.1 Logic Gates

So Starting with the power analysis of basic logic gates including AND, OR, NAND, NOR & various XOR design implementation.

7.2.1.1 AND Gate

Firstly when user looks forward to QCA Pro tool window, user have to choose a QCA cell layout of the design for which he wishes to estimate power calculation. Then he has to select input vectors for the design; they will be maintained in a text file showing all the possible combinations of input vectors as well as the no of total combinations, called as switching vector. Then another text files which having those all input vector combinations as well as their respected outputs, known as vector set. Then he has to choose an output directory in which all the processing data will be maintained. After that value of temperature and kink energy. Then user looks forward to checking the design, If it is verified then it will show a power tab within the window showing polarization map for the corresponding temperature and kink energy values. And if the output values don't match with expected one then it shows an error message. Fig. Shows primary window of QCA Pro tool :

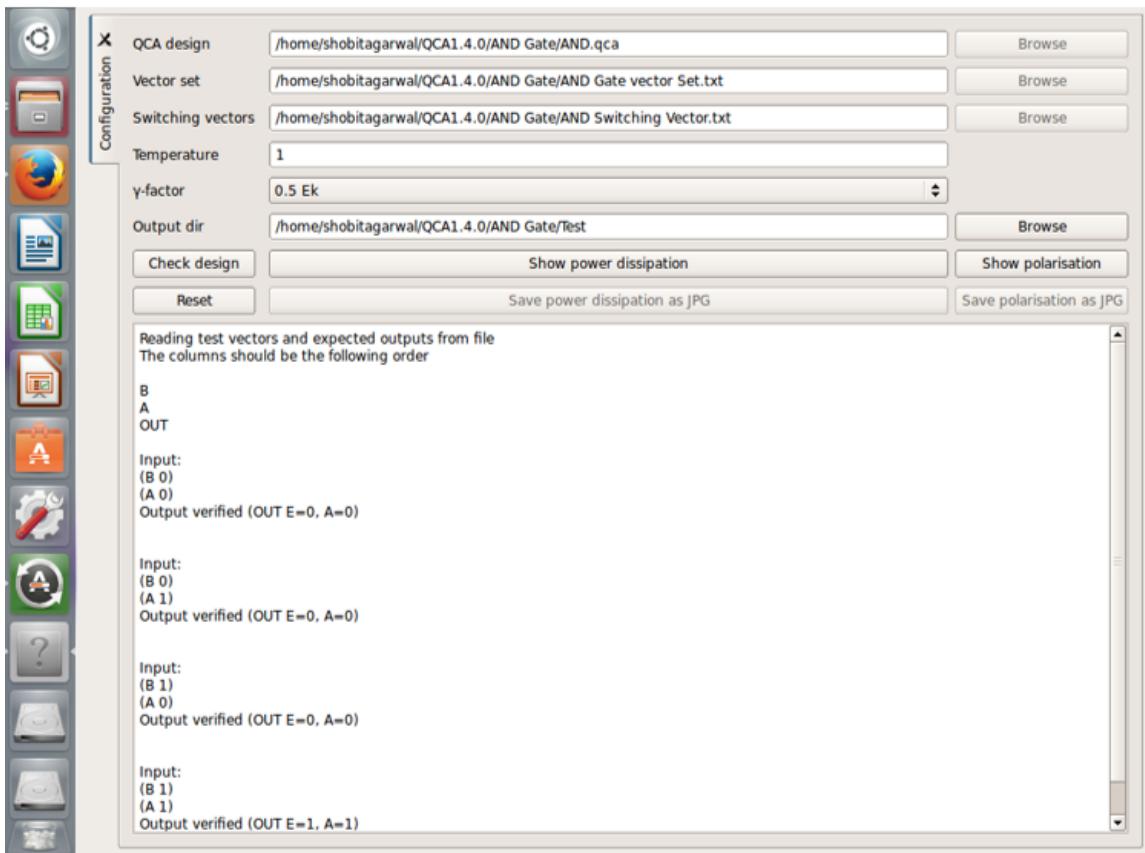


Figure 7.3: QCA Pro tool in action

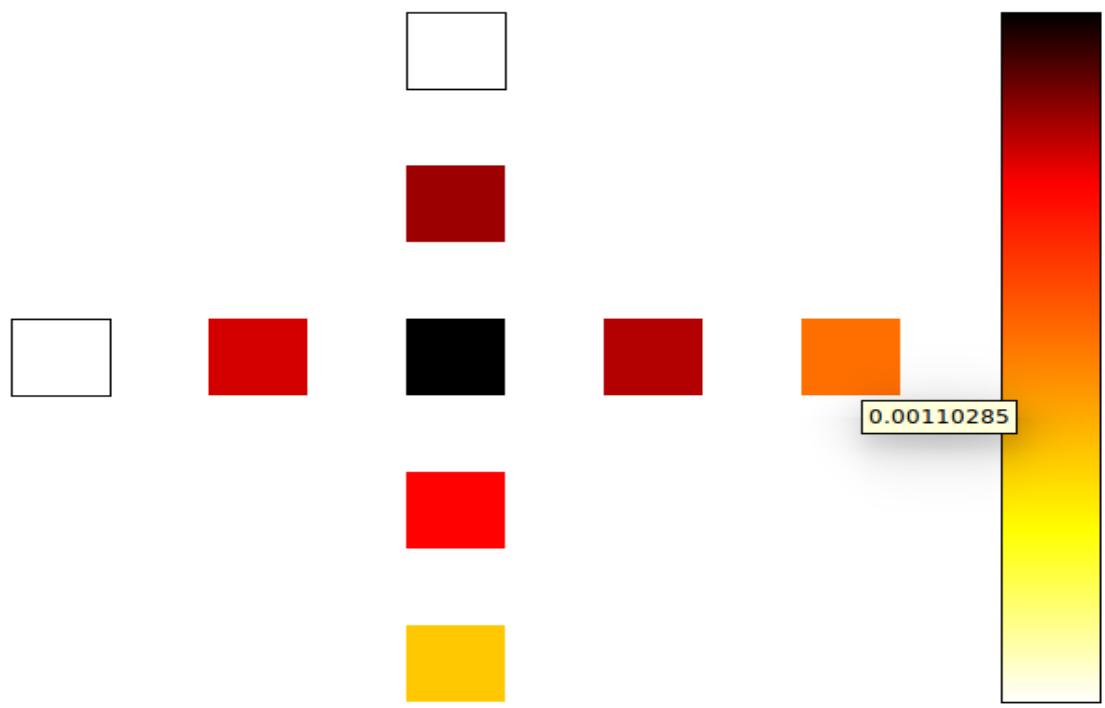


Figure 7.4: Power dissipation map for AND Gate (1, 0.5Ek)

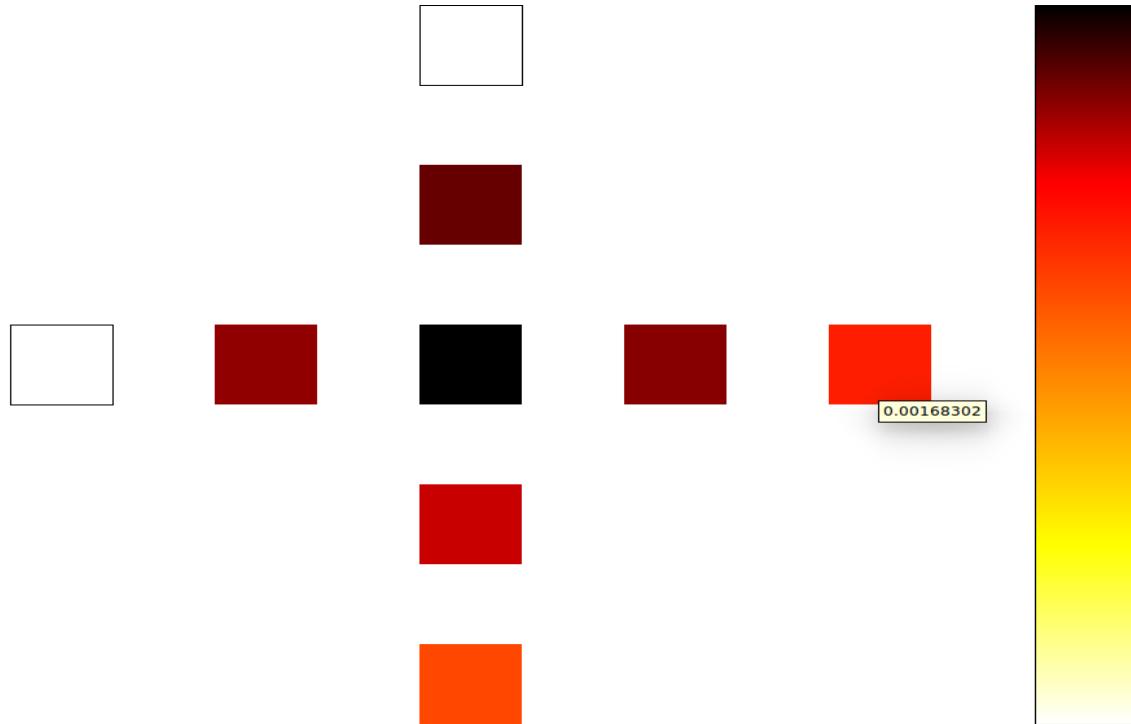


Figure 7.5: Power dissipation map for AND Gate (1, 1.0Ek)

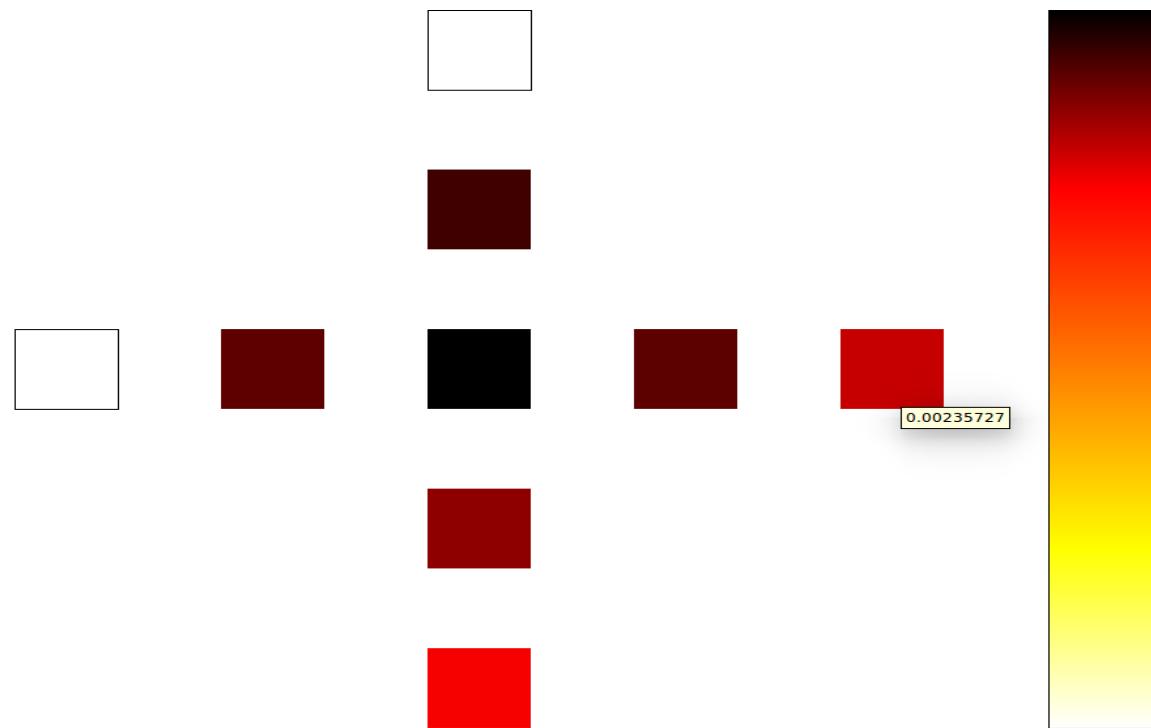


Figure 7.6: Power dissipation map for AND Gate (1, 1.5Ek)

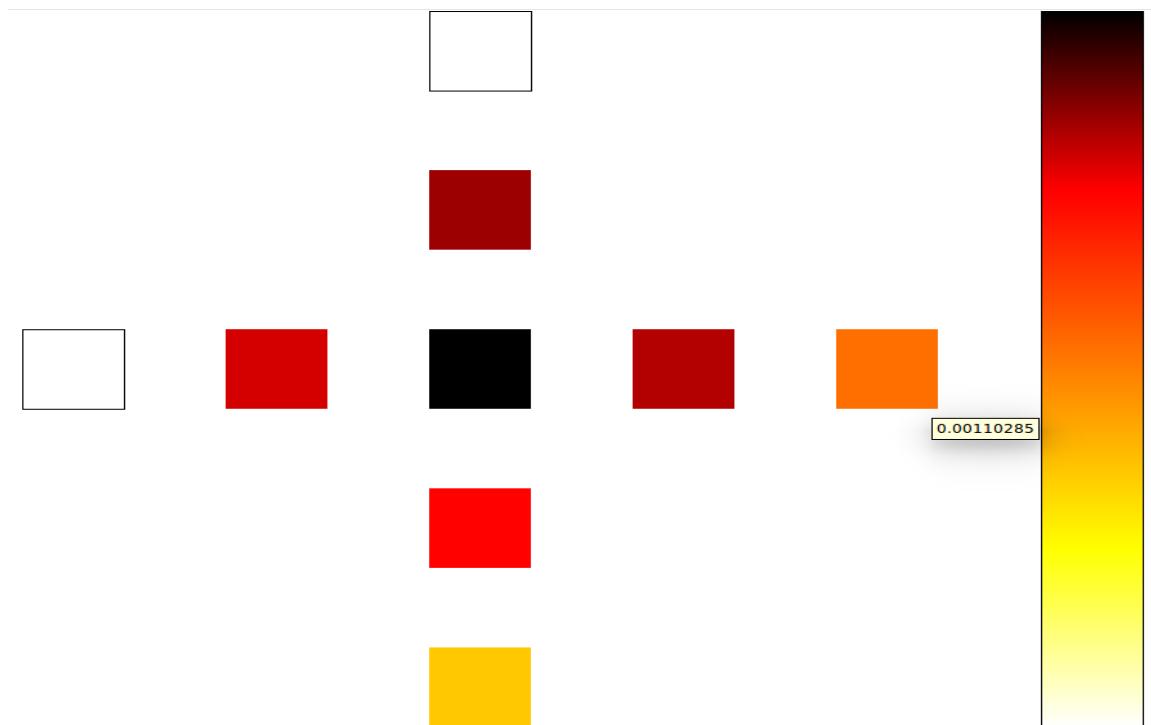


Figure 7.7: Power dissipation map for AND Gate (2, 0.5Ek)

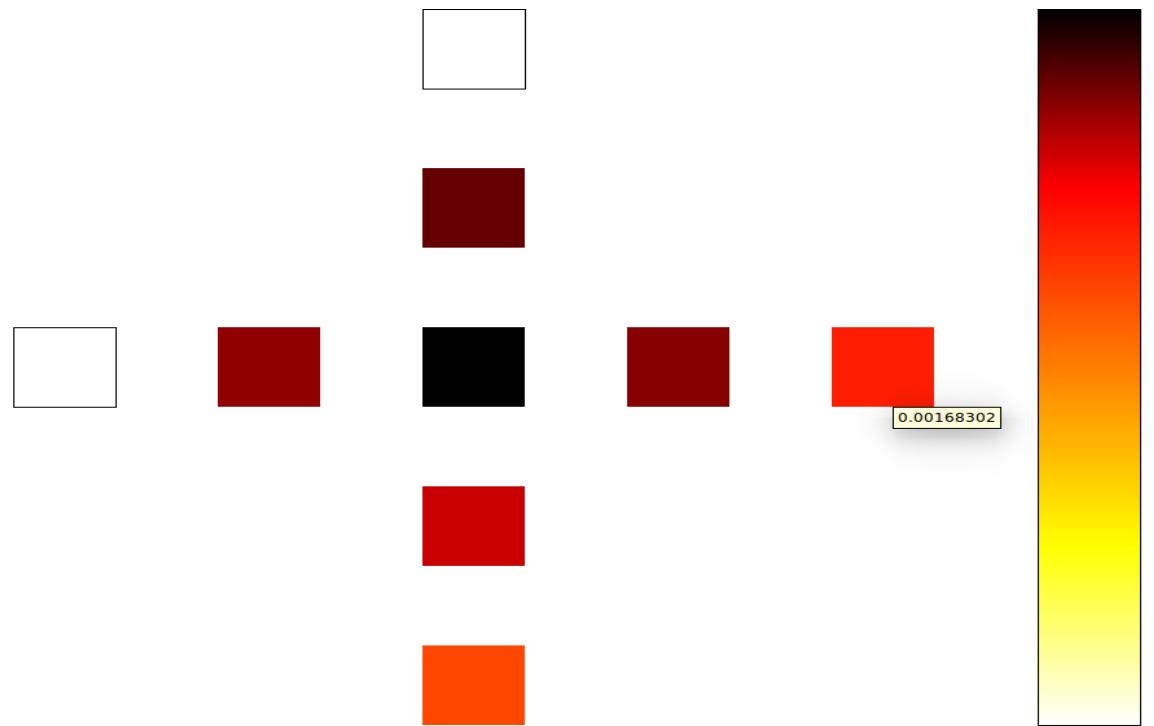


Figure 7.8: Power dissipation map for AND Gate (2, 1.0Ek)

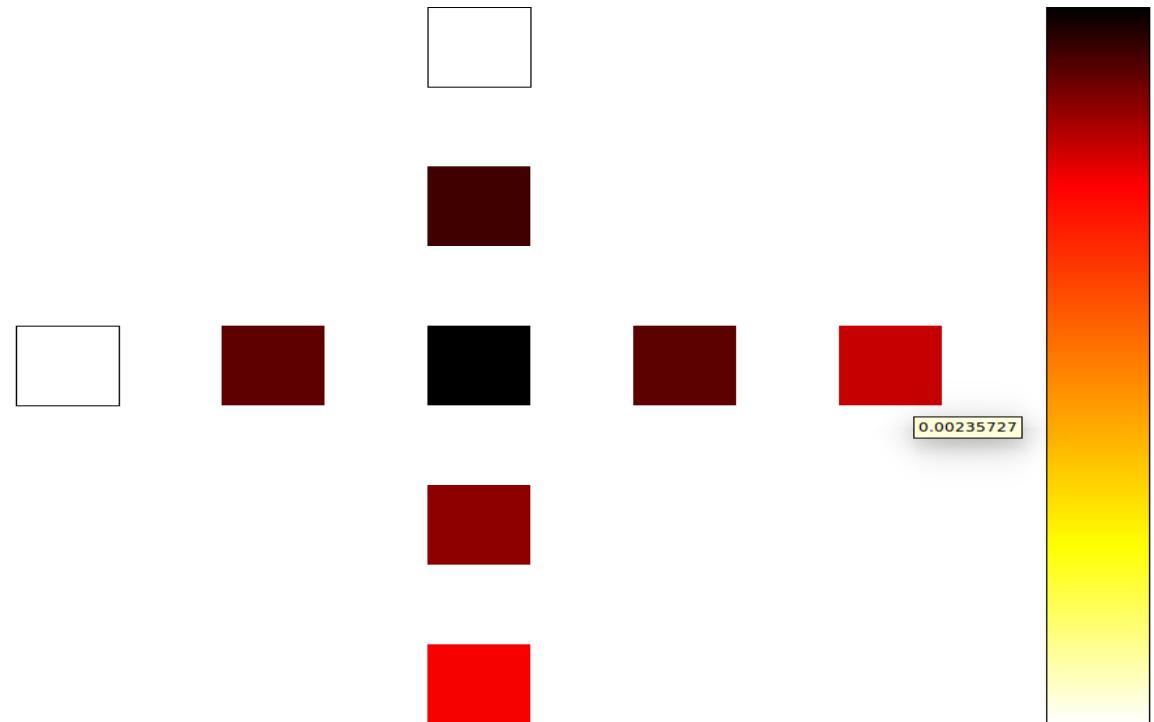


Figure 7.9: Power dissipation map for AND Gate (2, 1.5Ek)

Kink Energy	Tempearature	
	1	2
0.5	0.00110285	0.00110285
1.0	0.00168302	0.00168302
1.5	0.00235727	0.00235727

Table 7.1: Power dissipation values for AND gate

As shown above, all figures are having (Temperature, KinkEnergy) formatting in their name. And in the table, all data have been summarized.

7.2.1.2 OR Gate

All the steps as discussed earlier will be followed in similar fashion for power analysis using QCA Pro tool, But now onwards power dissipation map for all sets of temparature & kink energy is not going to be shown here as it will lead to increase the only number of pages, so only one power dissipation map is there for upcoming designs yet all the calculated data in tabular form is presented.

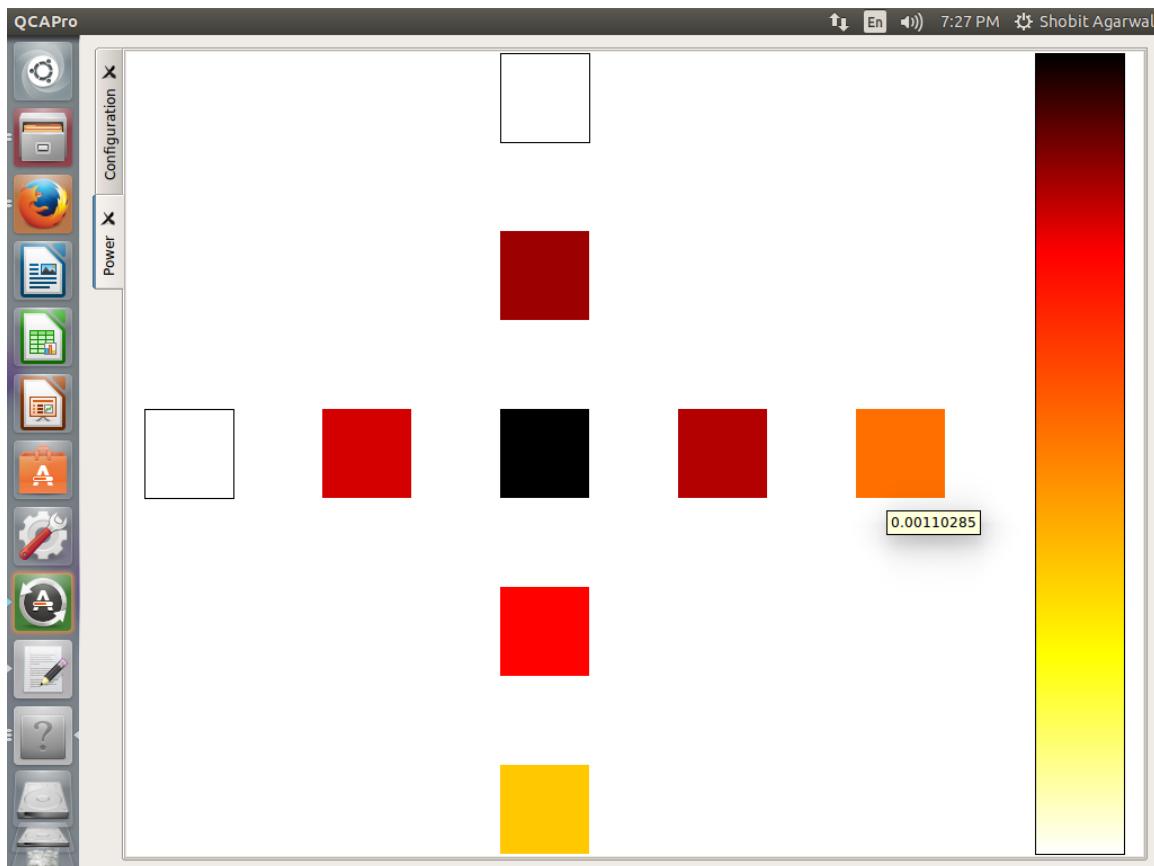


Figure 7.10: Power dissipation map for OR Gate (1, 0.5Ek)

Kink Energy	Tempearature	
	1	2
0.5	0.00110285	0.00110285
1.0	0.00168302	0.00168302
1.5	0.00235727	0.00235727

Table 7.2: Power dissipation values for OR gate

7.2.1.3 NAND Gate:

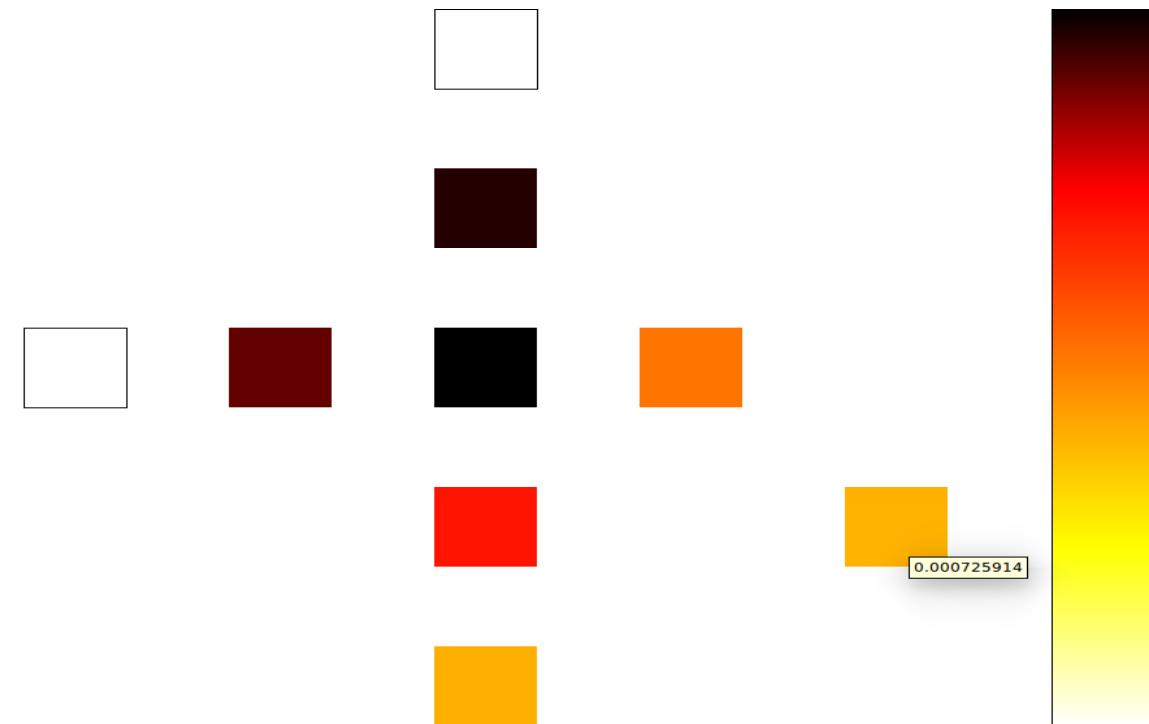


Figure 7.11: Power dissipation map for NAND Gate (1, 0.5Ek)

Kink Energy	Tempearature	
	1	2
0.5	0.000725914	0.000725844
1.0	0.00143955	0.00143895
1.5	0.00218447	0.00218343

Table 7.3: Power dissipation values for NAND gate

7.2.1.4 NOR Gate:

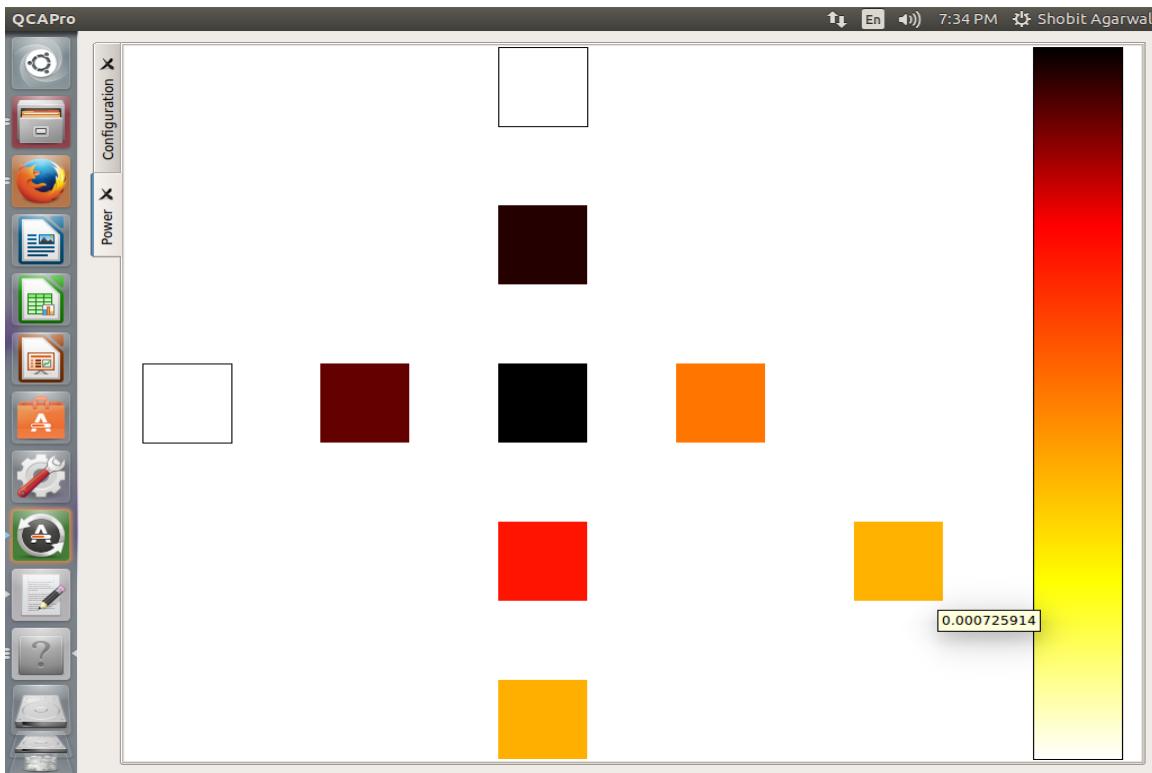


Figure 7.12: Power dissipation map for NOR Gate (1, 0.5Ek)

Kink Energy	Tempearature	
	1	2
0.5	0.000725914	0.000725844
1.0	0.00143955	0.00143895
1.5	0.00218447	0.00218343

Table 7.4: Power dissipation values for NOR gate

7.2.1.5 XOR Gate Design-1:

Since we have implemented XOR in various ways, hence here we present power analysis of all designs, one by one.

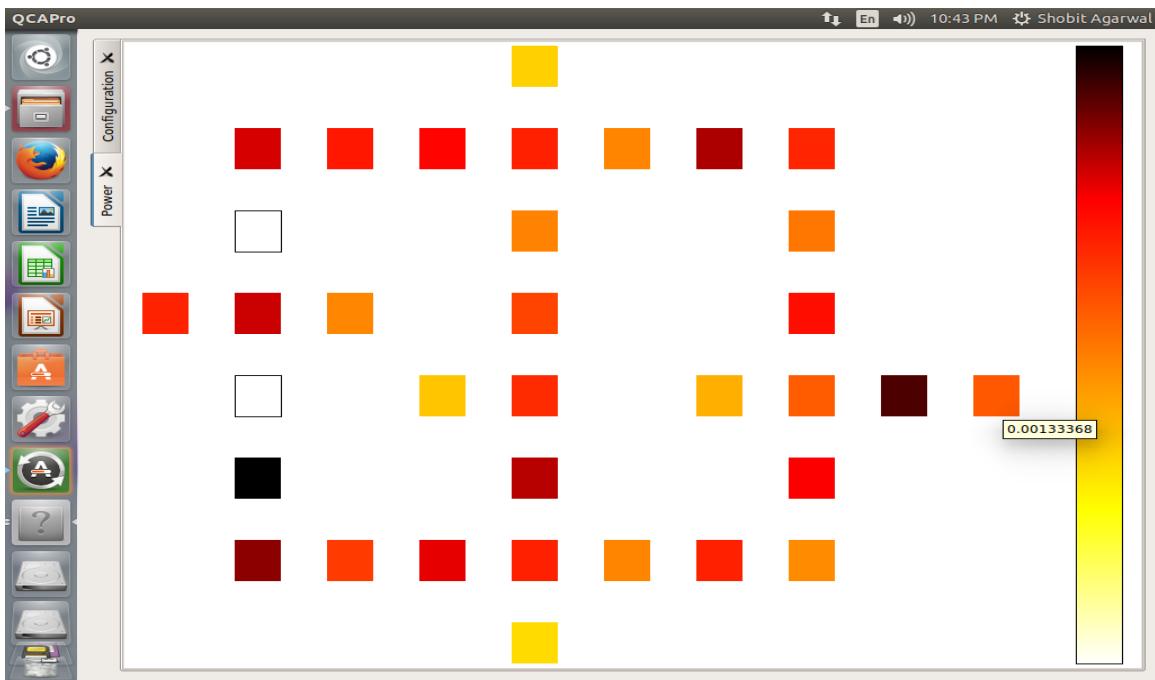


Figure 7.13: Power dissipation map for XOR Gate Design-1 (1, 0.5Ek)

7.2.1.6 XOR Gate Design-2:

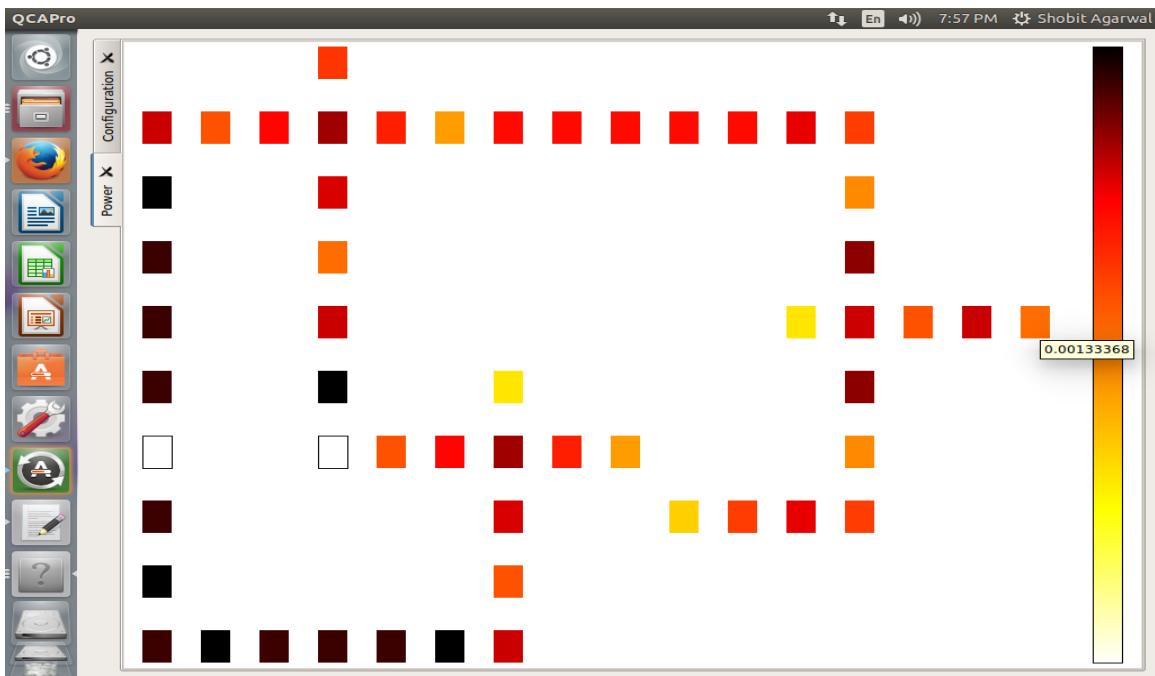


Figure 7.14: Power dissipation map for XOR Gate Design-2 (1, 0.5Ek)

7.2.1.7 XOR Gate Design-3:

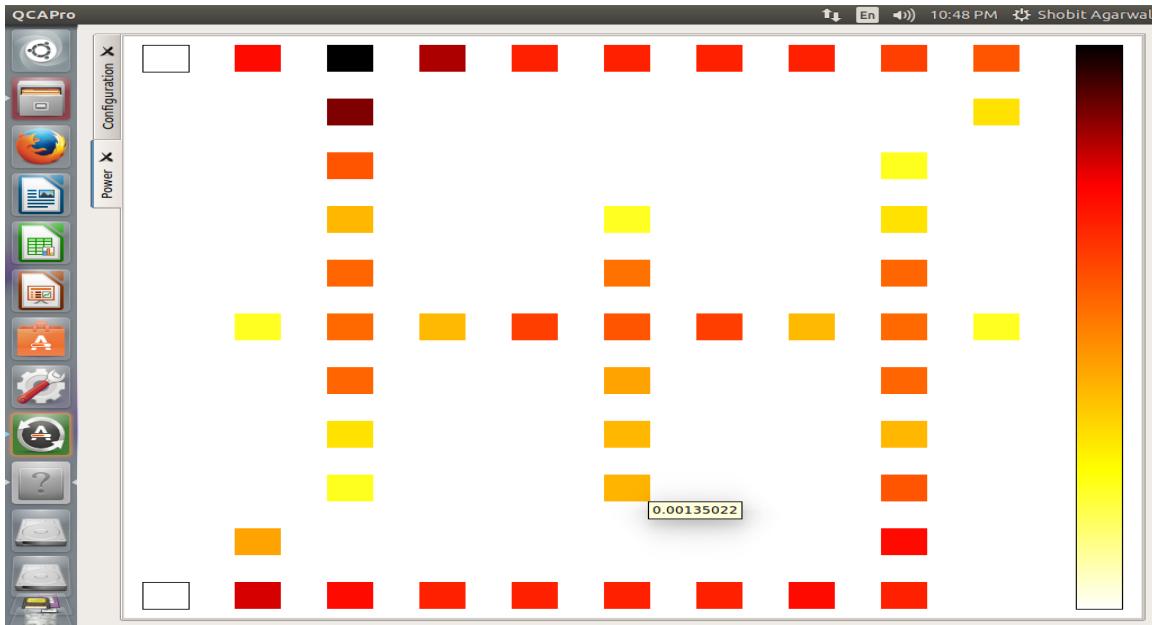


Figure 7.15: Power dissipation map for XOR Gate Design-3 (1, 0.5Ek)

7.2.1.8 XOR Gate Design-4:

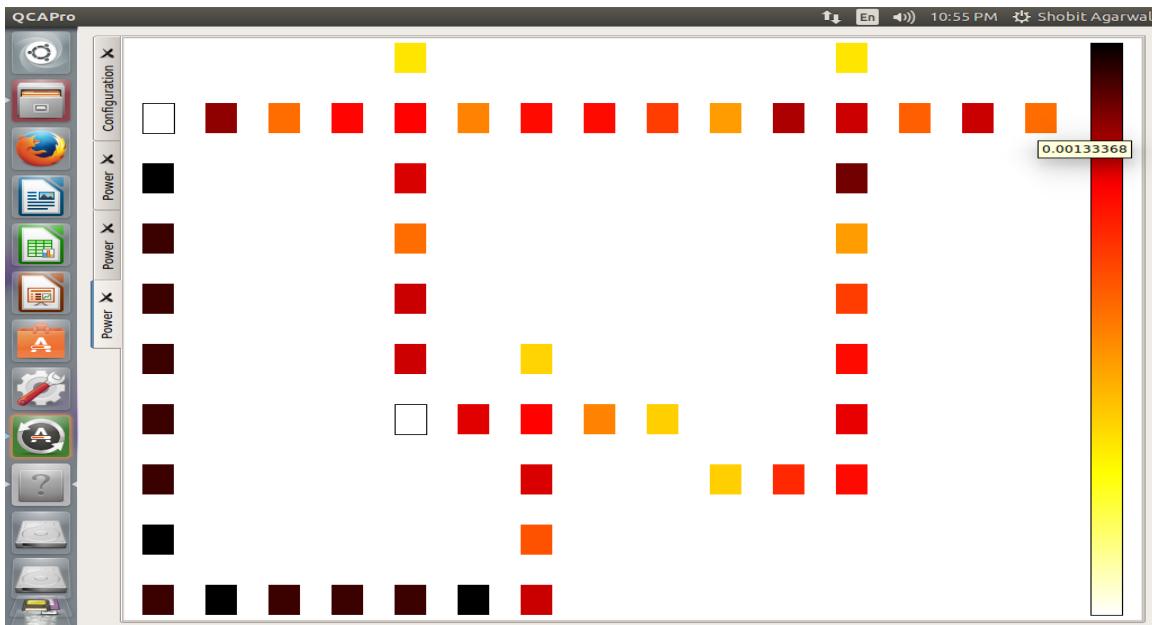


Figure 7.16: Power dissipation map for XOR Gate Design-4 (1, 0.5Ek)

7.2.1.9 XOR Gate Design-5:

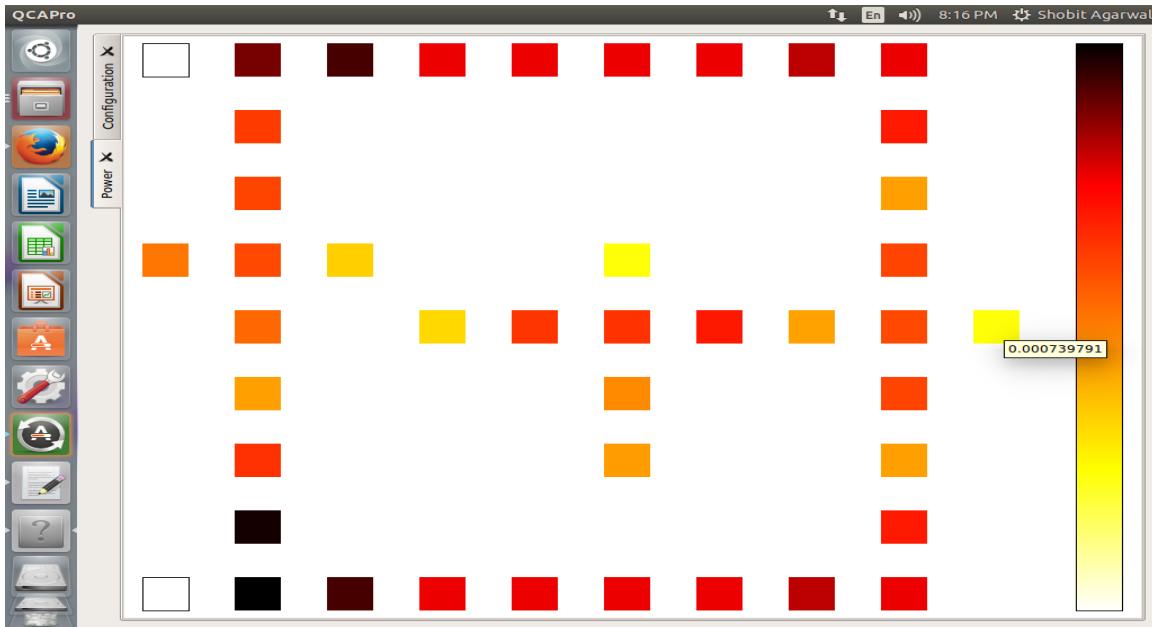


Figure 7.17: Power dissipation map for XOR Gate Design-5 (1, 0.5Ek)

7.2.1.10 XOR Gate Design-6:

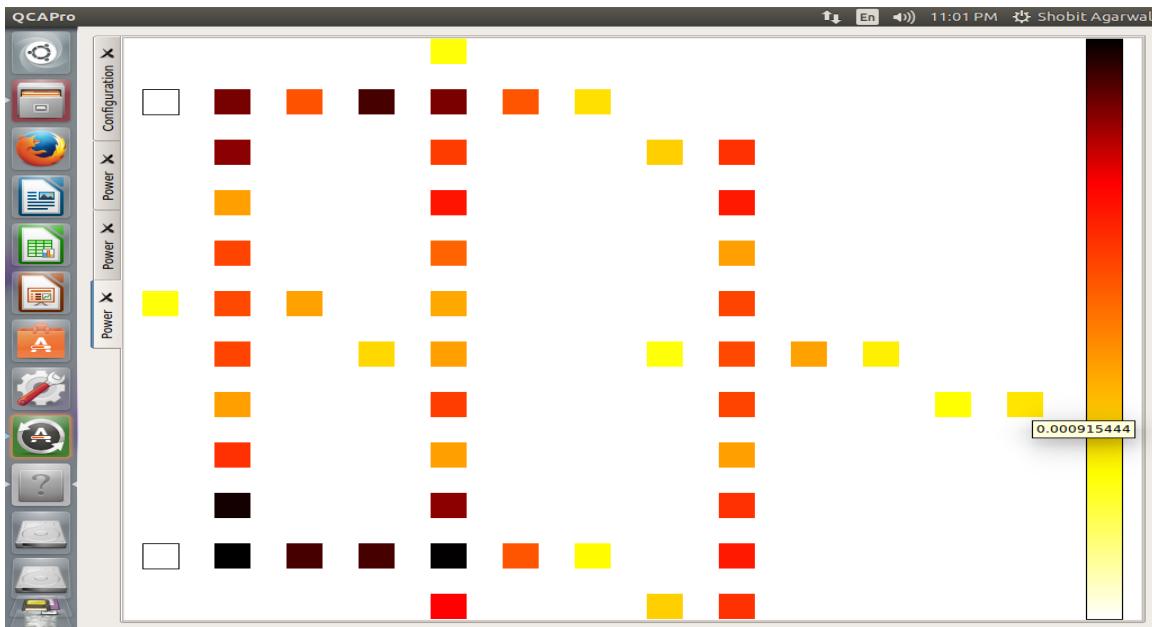


Figure 7.18: Power dissipation map for XOR Gate Design-6 (1, 0.5Ek)

7.2.1.11 XOR Gate Design-7:

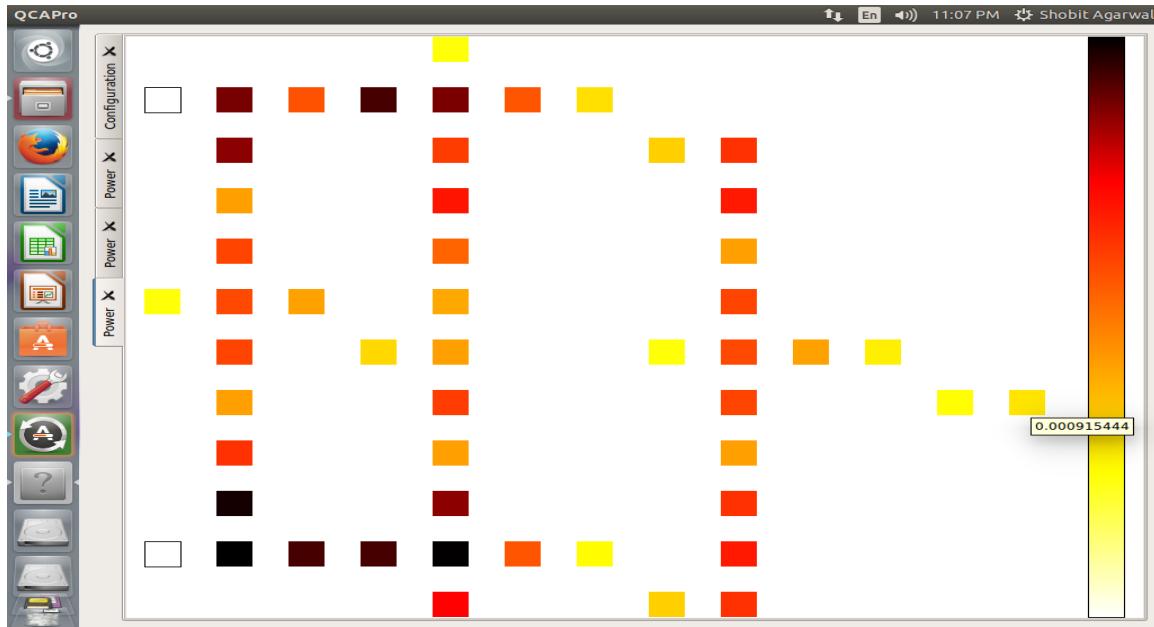


Figure 7.19: Power dissipation map for XOR Gate Design-7 (1, 0.5Ek)

7.2.1.12 XOR Gate Design-8:

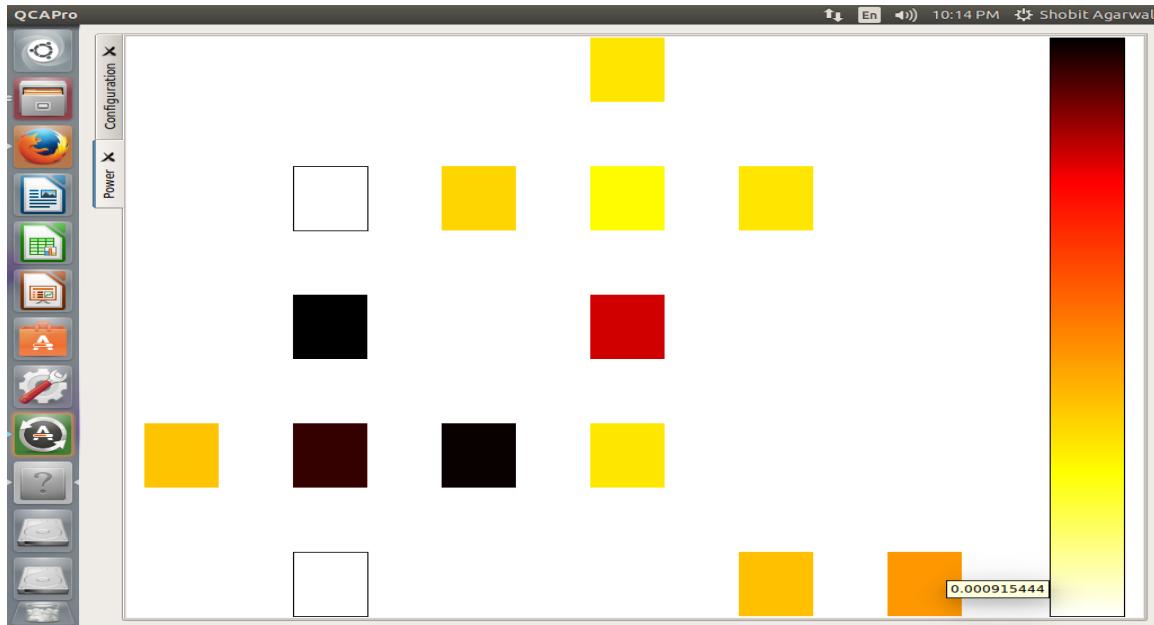


Figure 7.20: Power dissipation map for XOR Gate Design-8 (1, 0.5Ek)

7.2.1.13 Performance Analysis of XOR Gate Designs:

Since there are 8 designs for XOR Gate, thus we have analyzed them regarding power dissipation. Tables for Avg. Leakage Energy Dissipation and Avg. Switching Energy Dissipation is shown below.

	0.5 E_k	1.0 E_k	1.5 E_k
XOR8	05.68	14	22.87
XOR1	12.02	34.36	59.17
XOR5	12.46	38.33	69.14
XOR3	15.85	46.20	81.58
XOR4	15.53	47.26	84.49
XOR2	15.57	48.80	88.30
XOR6	18.06	51.26	88.99
XOR7	18.06	51.26	88.99

Table 7.5: Avg. Leakage Energy dissipation (mev) at $T=1^0K$

	1.5 E_k	1.0 E_k	0.5 E_k
XOR8	05.59	06.69	07.30
XOR1	23.84	29.18	35.51
XOR5	50.74	59.76	68.84
XOR6	50.68	60.09	70.41
XOR7	50.68	60.09	70.41
XOR3	52.81	62.35	72.26
XOR4	52.30	62.27	72.70
XOR2	57.54	68.41	79.65

Table 7.6: Avg. Switching Energy dissipation (mev) at $T=1^0K$

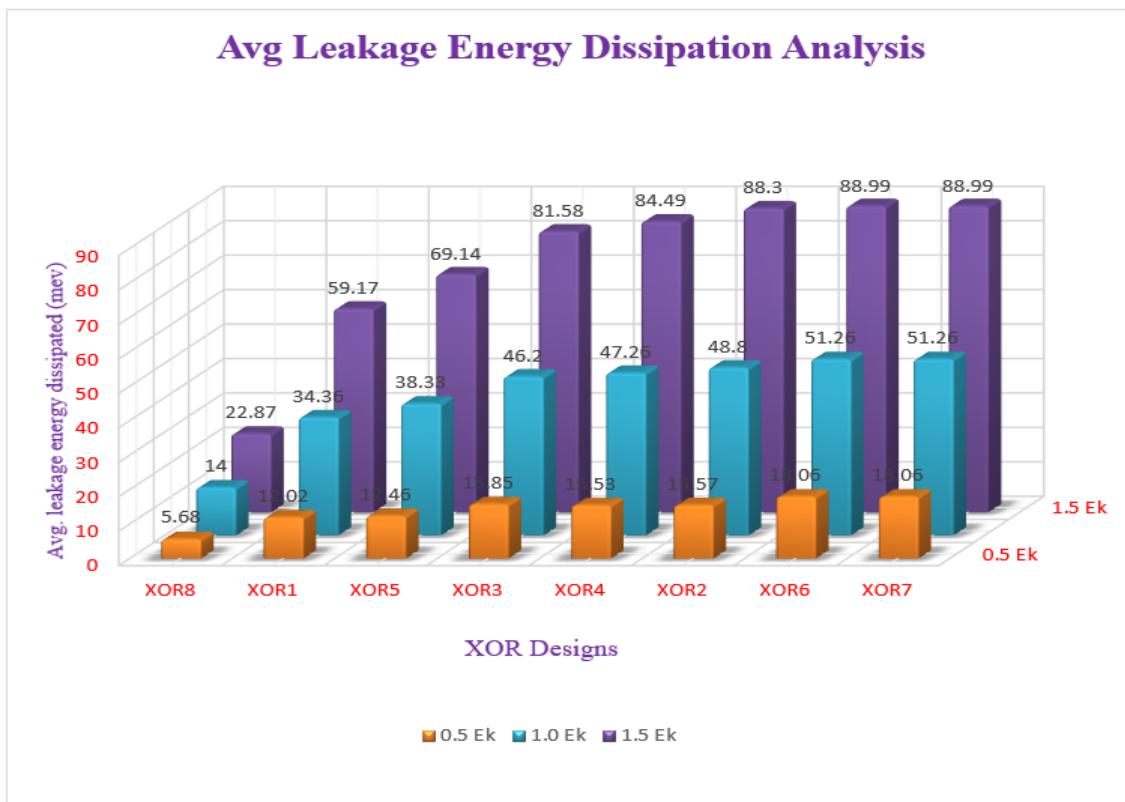


Figure 7.21: Avg. Leakage Energy dissipation (mev) for XOR Gate designs at $T=1^0K$

	0.5 E_k	1.0 E_k	1.5 E_k
XOR8	12.98	20.69	28.83
XOR1	47.53	63.54	83.01
XOR5	81.29	98.09	119.88
XOR3	88.11	108.56	134.39
XOR4	88.24	109.52	136.79
XOR6	88.47	111.35	139.67
XOR7	88.47	111.35	139.67
XOR2	95.22	117.21	145.83

Table 7.7: Avg. Energy dissipation in circuit (mev) at $T=1^0K$

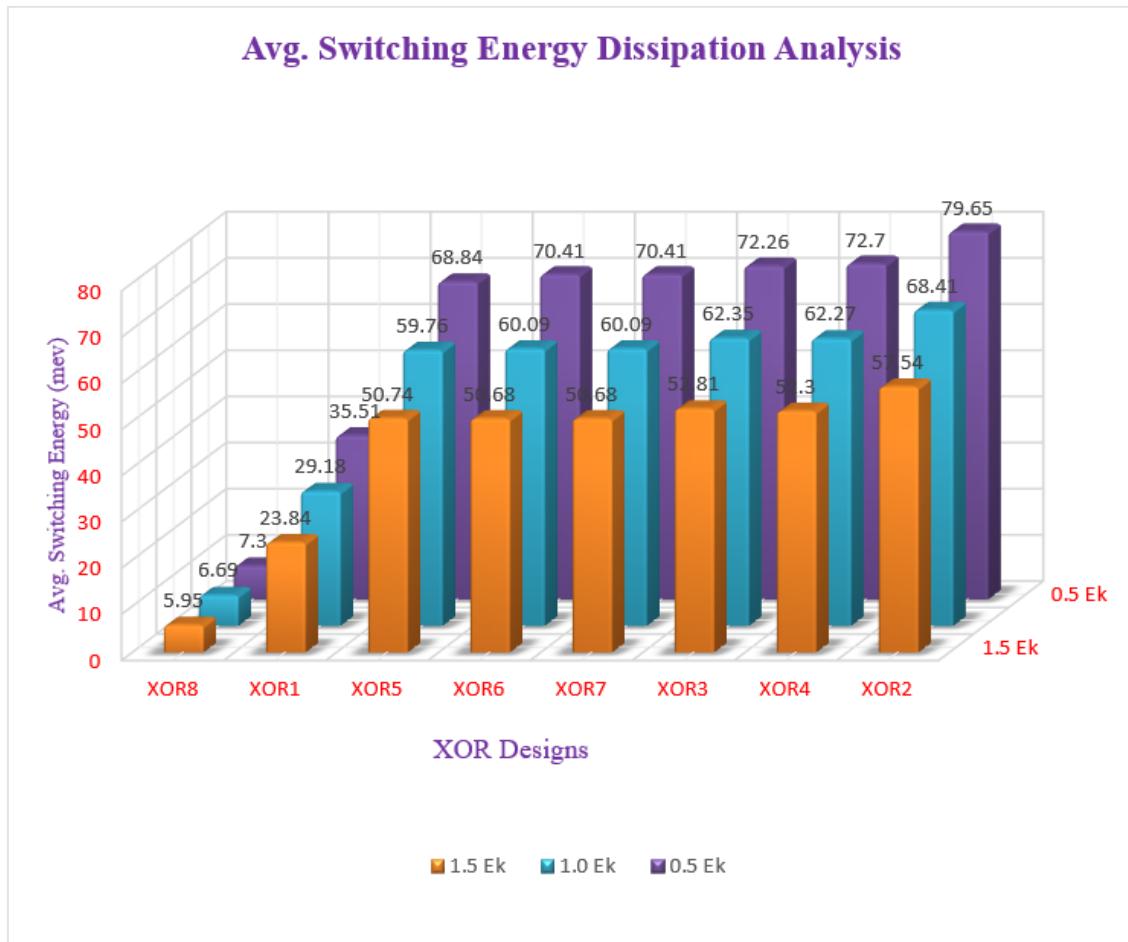


Figure 7.22: Avg. Switching Energy dissipation (mev) for XOR Gate designs at $T=1^0K$

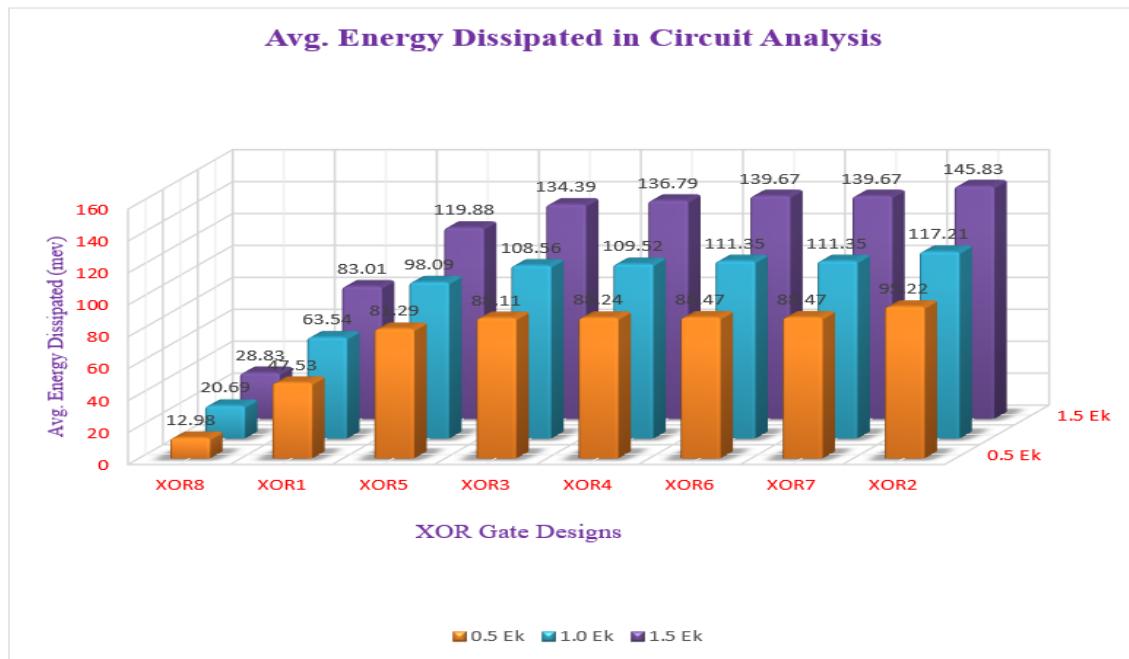


Figure 7.23: Avg. Energy dissipation in circuit (mev) for XOR Gate designs at $T=1^0K$

7.2.2 MUX Designs

7.2.2.1 MUX Design-1

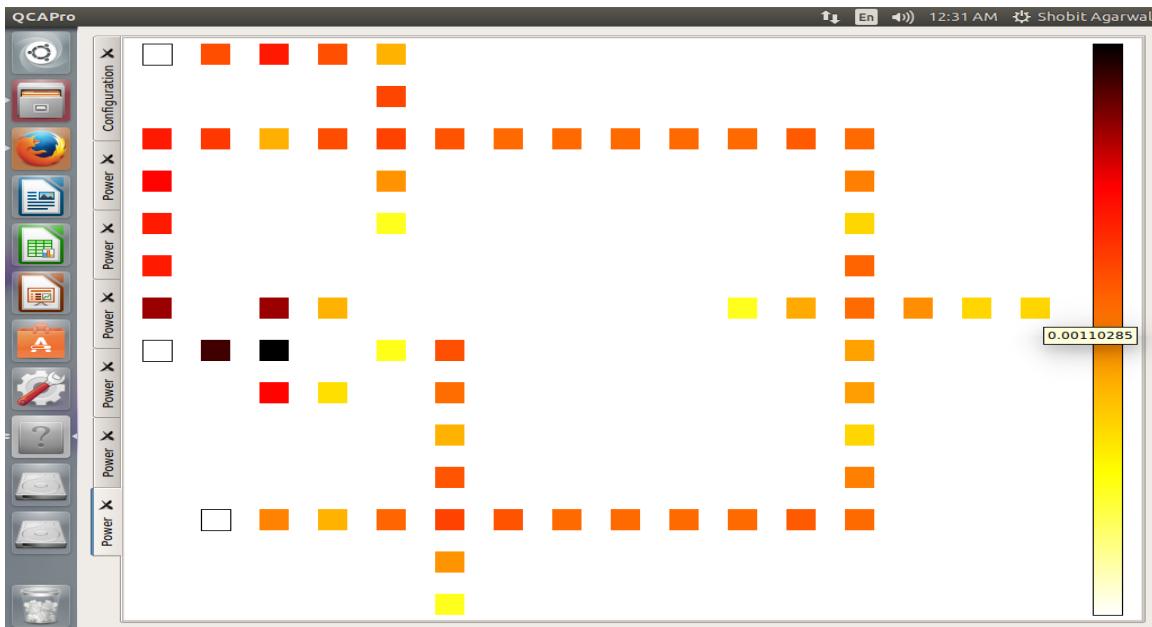


Figure 7.24: Power dissipation map for MUX Design-1 (1, 0.5Ek)

7.2.2.2 MUX Design-2

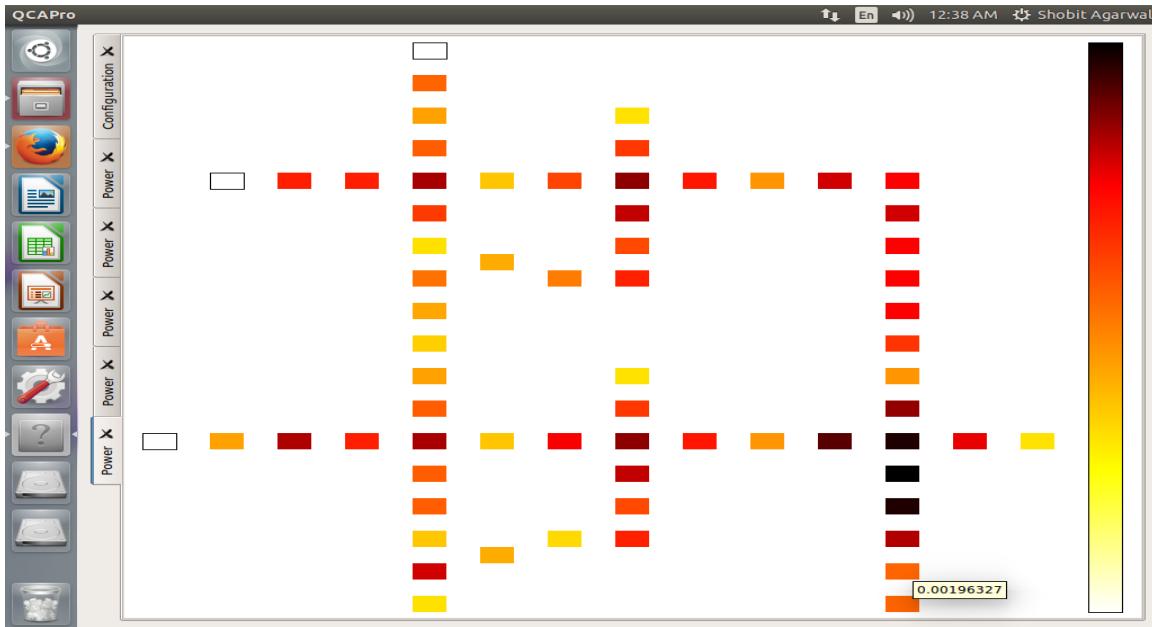


Figure 7.25: Power dissipation map for MUX Design-2 (1, 0.5Ek)

7.2.2.3 MUX Design-3

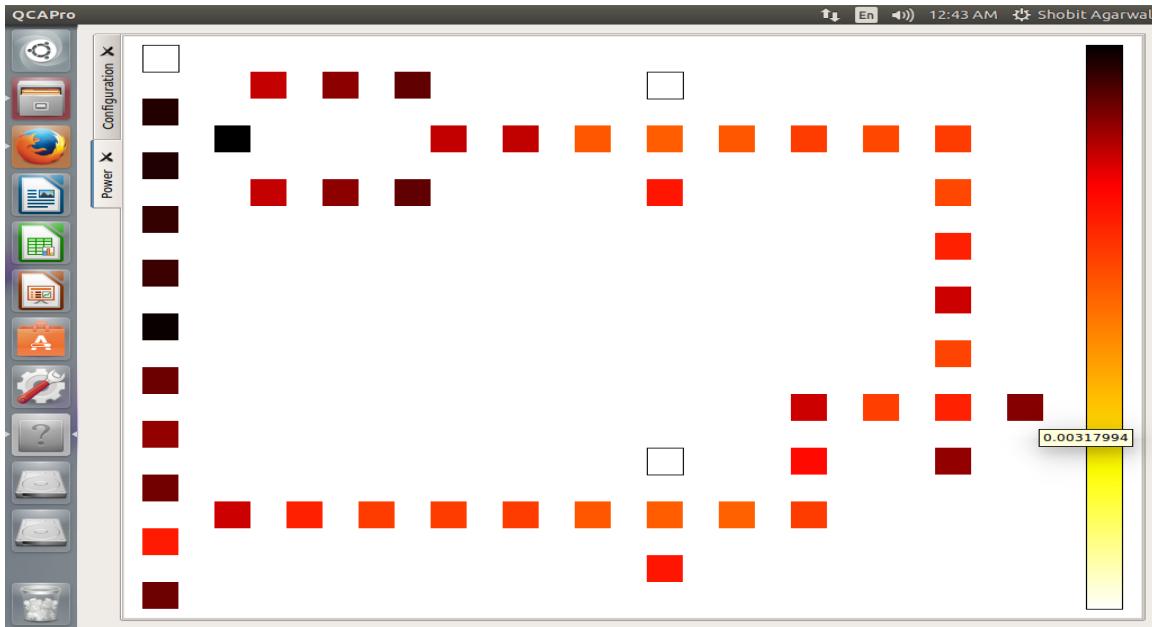


Figure 7.26: Power dissipation map for MUX Design-3 (1, 0.5Ek)

7.2.2.4 MUX Design-4

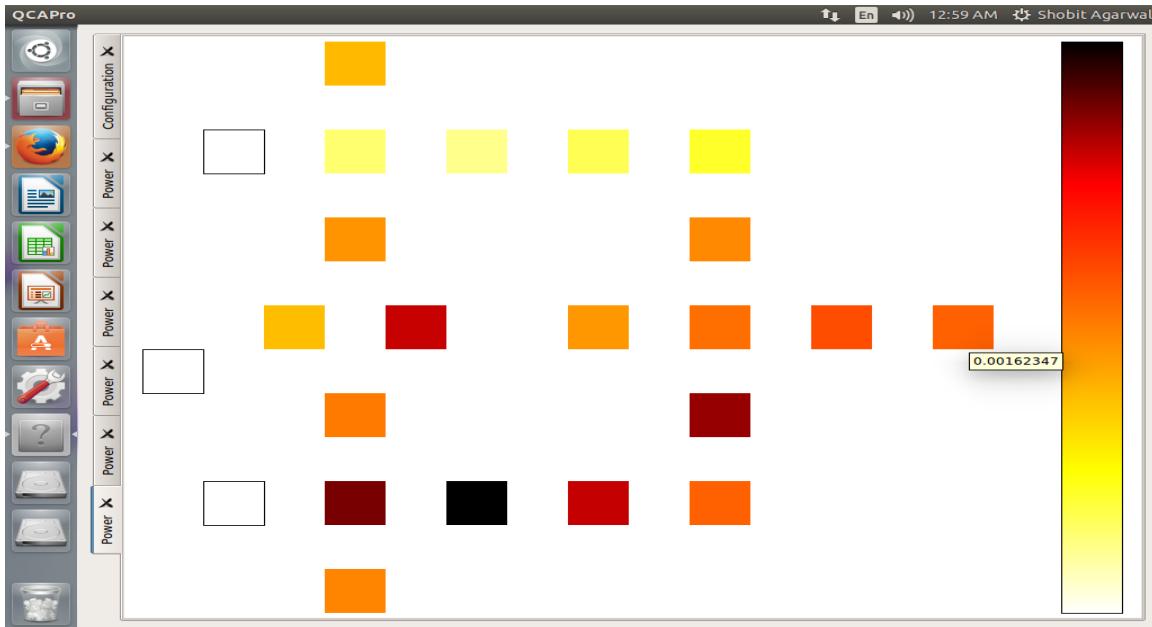


Figure 7.27: Power dissipation map for MUX Design-4 (1, 0.5Ek)

7.2.2.5 MUX Design-5

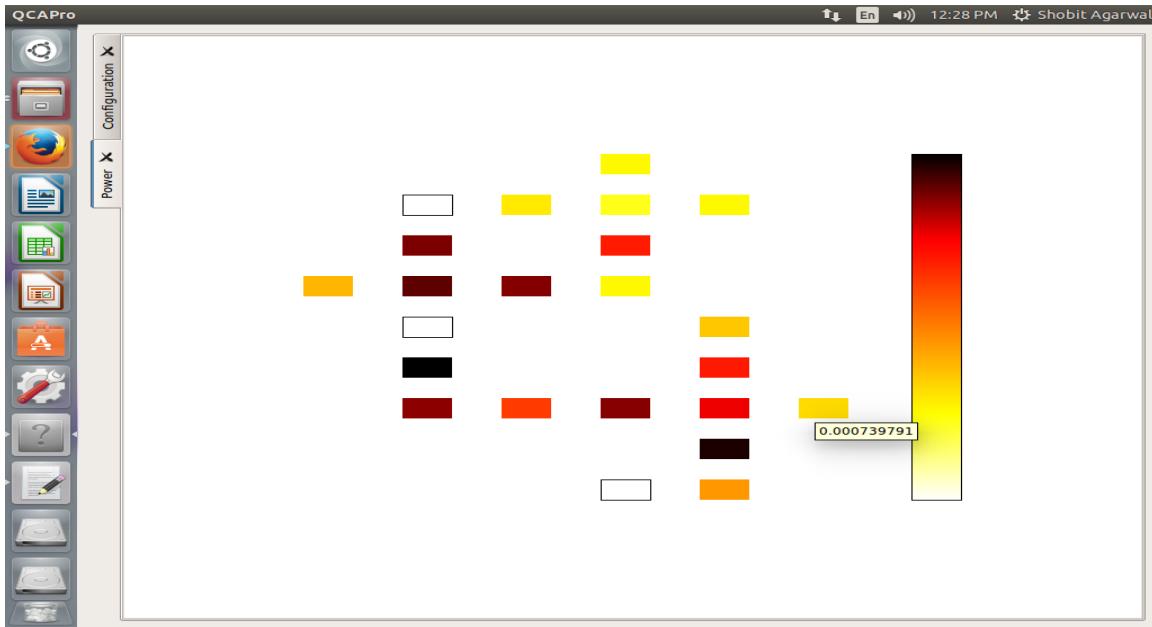


Figure 7.28: Power dissipation map for MUX Design-5 (1, 0.5Ek)

7.2.2.6 Performance Analysis of MUX Designs

	0.5 E_k	1.0 E_k	1.5 E_k
MUX 5	08.35	22.07	37.09
MUX 4	12.36	33.15	55.86
MUX 1	18.92	57.50	103.32
MUX 3	25.37	71.30	122.71
MUX 2	37.44	102.48	174.57

Table 7.8: Avg. Leakage Energy dissipation (mev) at $T=1^0K$

	1.5 E_k	1.0 E_k	0.5 E_k
MUX 3	04.72	05.96	08.08
MUX 4	11.42	14.12	17.66
MUX 5	14.30	16.75	19.32
MUX 1	65.97	77.80	89.93
MUX 2	70.43	85.53	103.51

Table 7.9: Avg. Switching Energy dissipation (mev) at $T=1^0K$

	0.5 E_k	1.0 E_k	1.5 E_k
MUX 3	27.66	38.82	51.39
MUX 4	30.01	47.27	67.28
MUX 5	33.46	77.26	127.42
MUX 1	108.85	135.29	169.29
MUX 2	140.95	188	245

Table 7.10: Avg. Energy dissipation in circuit (mev) at $T=1^0K$

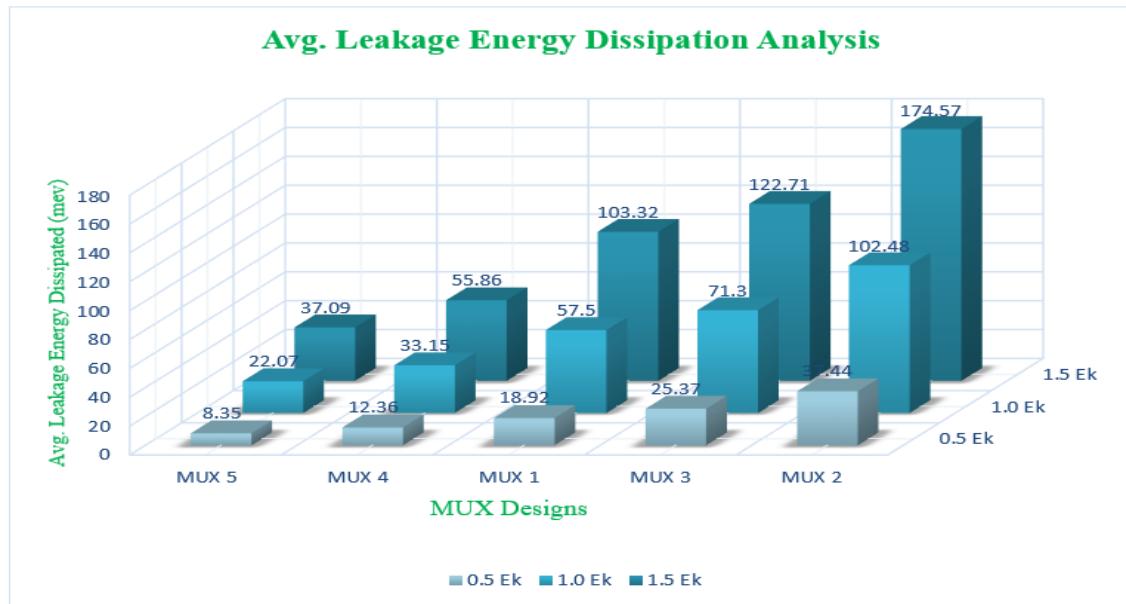


Figure 7.29: Avg. Leakage Energy dissipation (mev) for MUX designs at $T=1^0K$

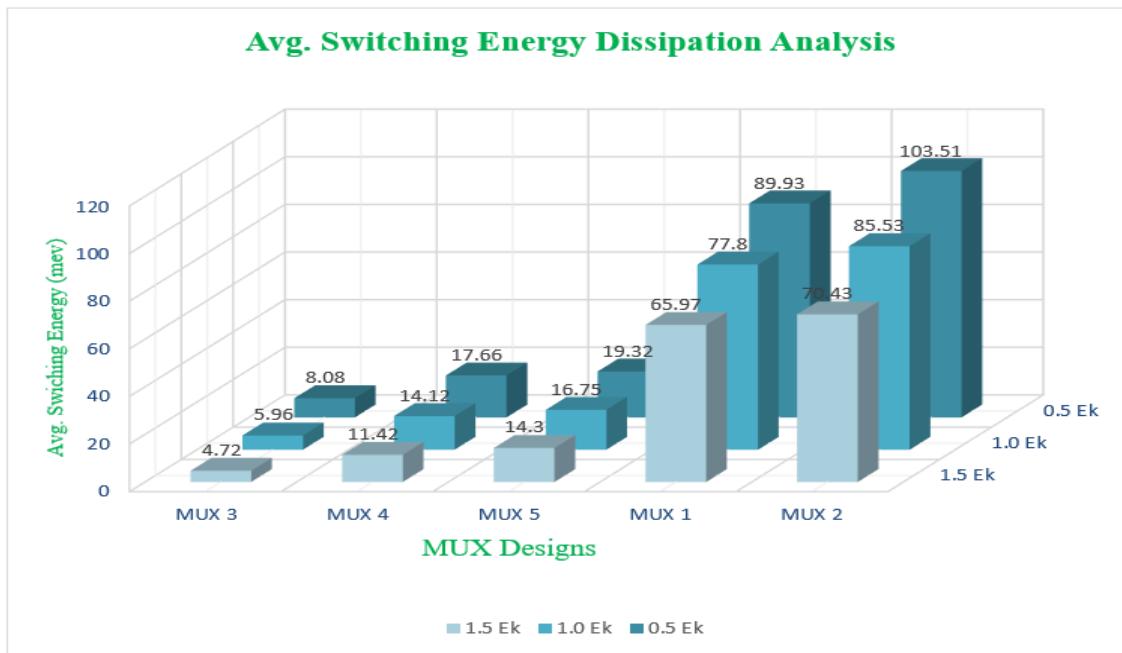


Figure 7.30: Avg. Switching Energy dissipation (mev) for MUX designs at $T=1^0K$

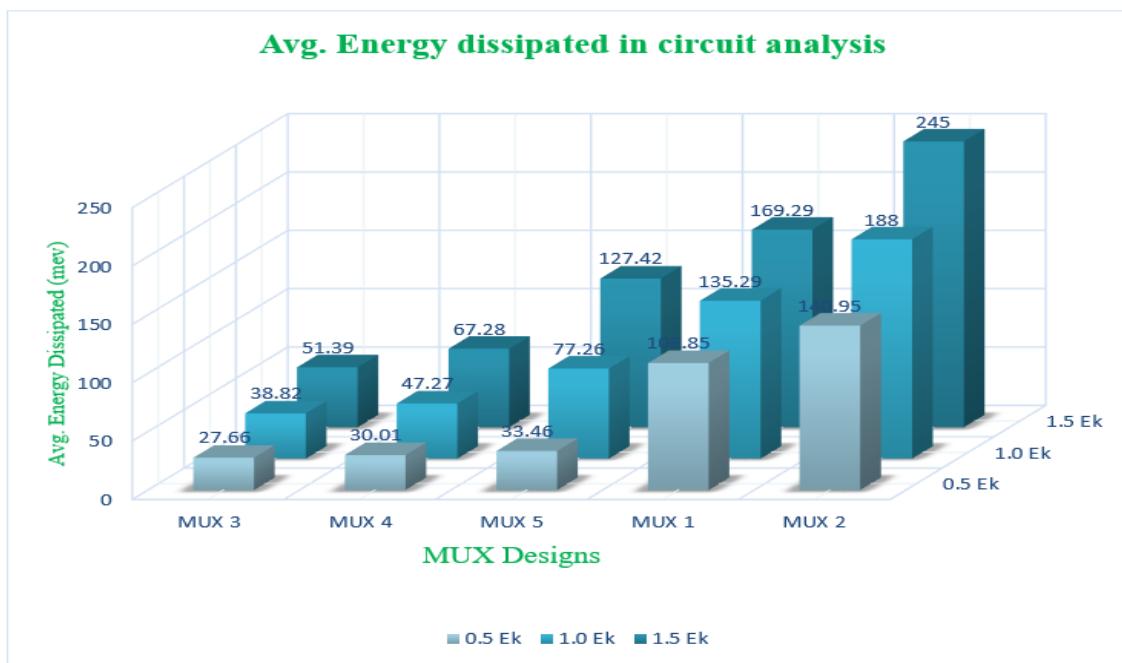


Figure 7.31: Avg. Energy dissipation in circuit (mev) for MUX designs at $T=1^0K$

7.2.3 Adder Designs

7.2.3.1 Design-1

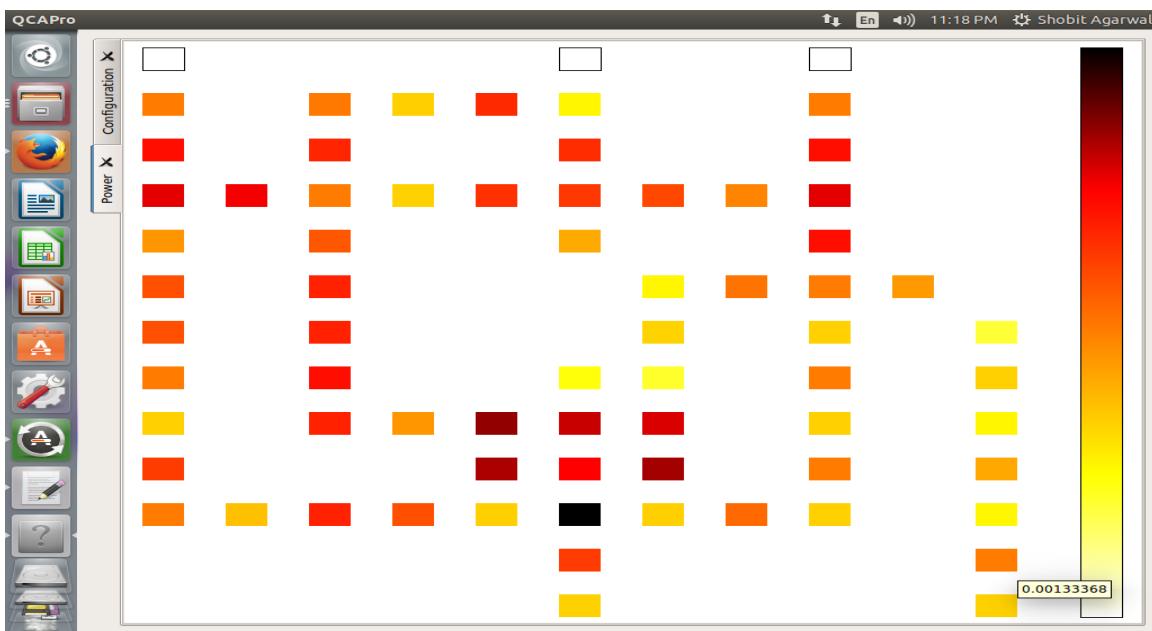


Figure 7.32: Power dissipation map for Carry of Design-1 (1, 0.5Ek)

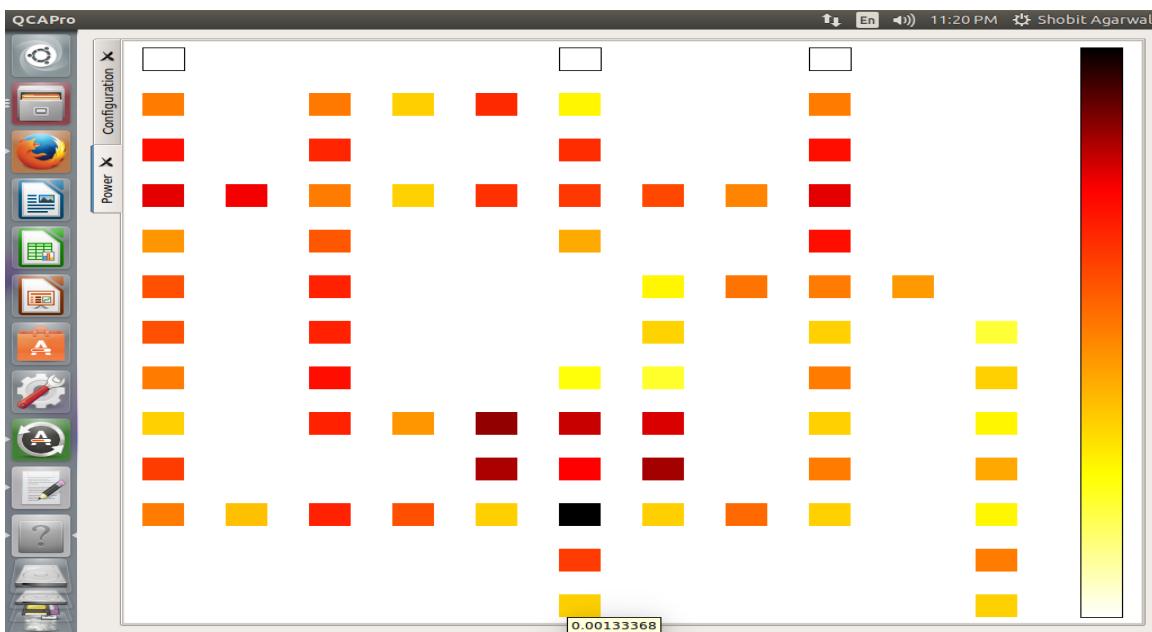


Figure 7.33: Power dissipation map for SUM of Design-1 (1, 0.5Ek)

7.2.3.2 Design-2

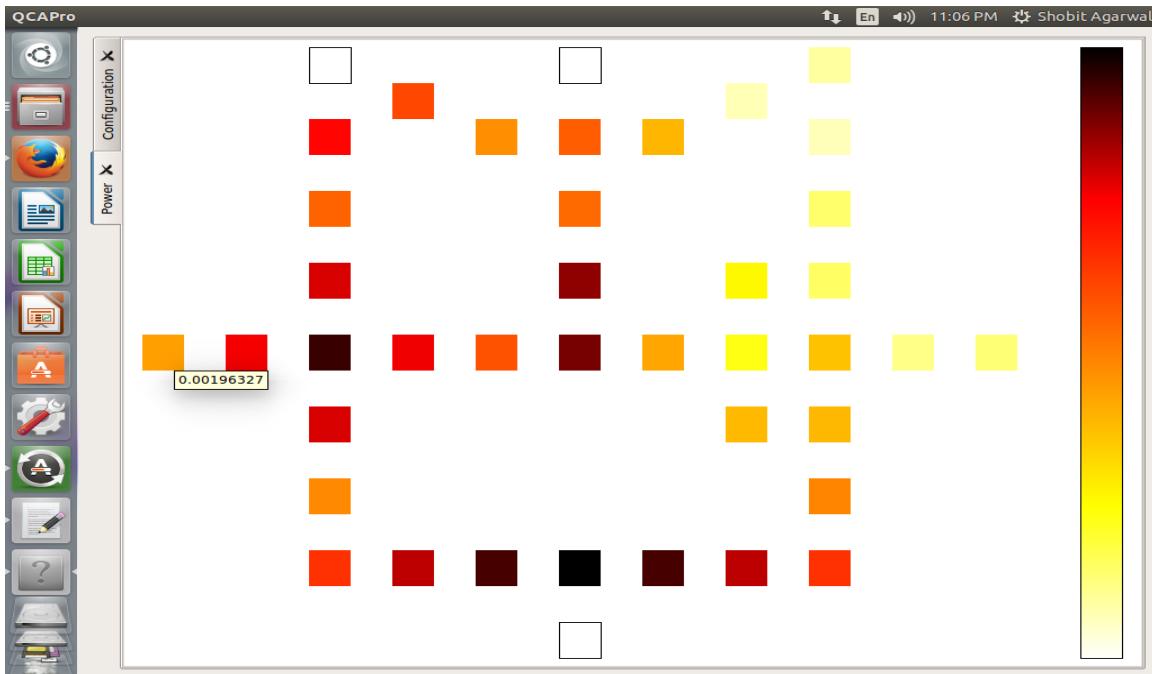


Figure 7.34: Power dissipation map for Carry of Design-2 (1, 0.5Ek)

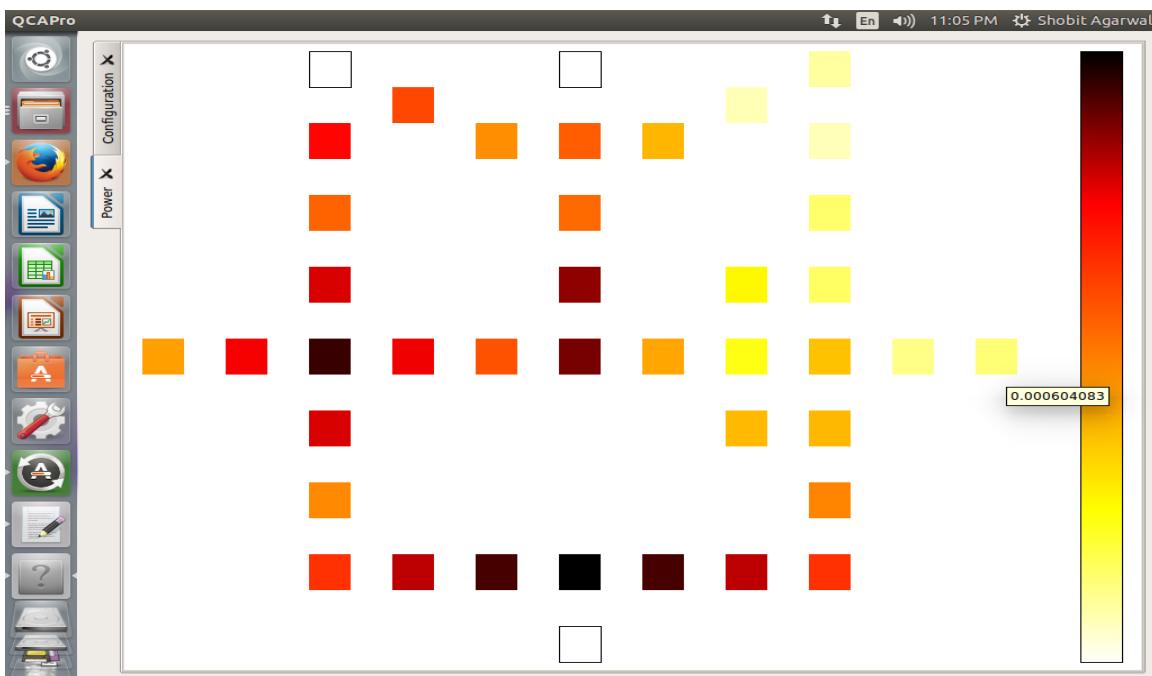


Figure 7.35: Power dissipation map for SUM of Design-2 (1, 0.5Ek)

7.2.3.3 Design-3



Figure 7.36: Power dissipation map for Carry of Design-3 (1, 0.5Ek)

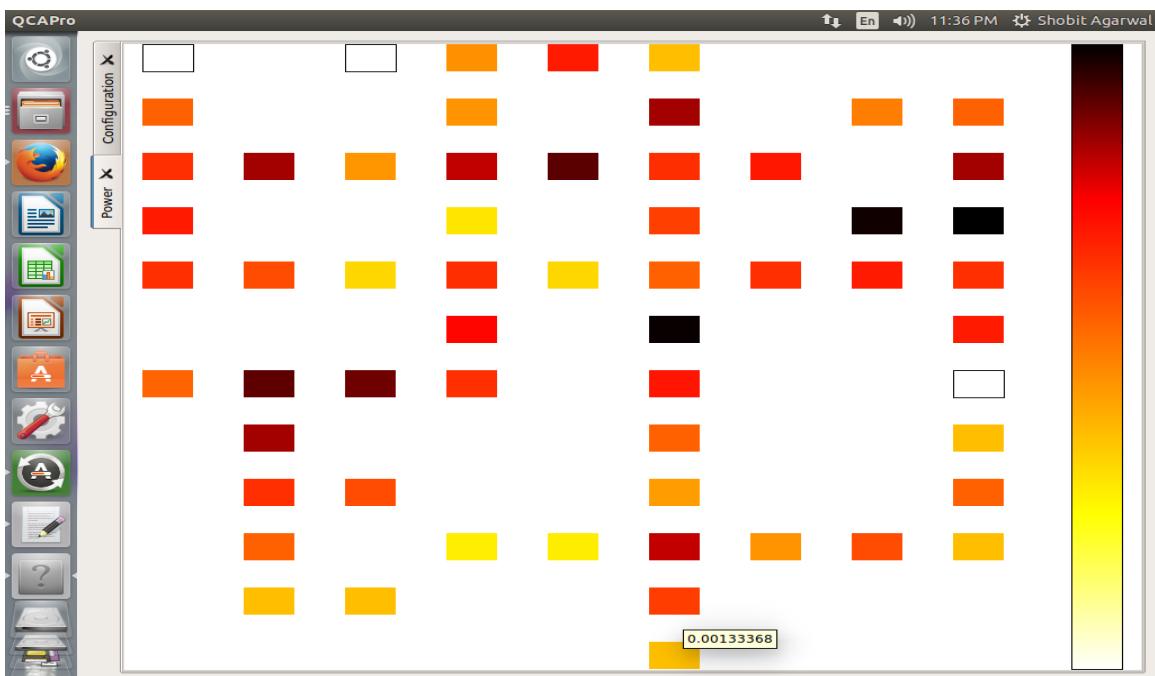


Figure 7.37: Power dissipation map for SUM of Design-3 (1, 0.5Ek)

7.2.3.4 Performance Analysis of Adder Desings

	0.5 E_k	1.0 E_k	1.5 E_k
Proposed Modified	15.77	49.88	90.43
Design 3	16.43	51.09	92.60
Proposed Basic	16.73	52.04	93.06
Design 2	18.37	54.76	96.96
Design 1	21.52	64.82	115.34

Table 7.11: Avg. Leakage Energy dissipation (mev) at $T=1^0K$

	1.5 E_k	1.0 E_k	0.5 E_k
Proposed Modified	47.22	59.09	70.22
Design 2	53.01	62.49	72.71
Proposed Basic	53.77	64.23	75.07
Design 3	79.61	92.68	105.94
Design 1	90.81	105.62	121.40

Table 7.12: Avg. Switching Energy dissipation (mev) at $T=1^0K$

	0.5 E_k	1.0 E_k	1.5 E_k
Proposed Modified	89.98	116.03	145
Design 2	91.08	117.25	149.96
Proposed Basic	93.27	118.12	150.28
Design 3	122.37	143.77	172.20
Design 1	142.91	170.44	206.15

Table 7.13: Avg. Energy dissipation in circuit (mev) at $T=1^0K$

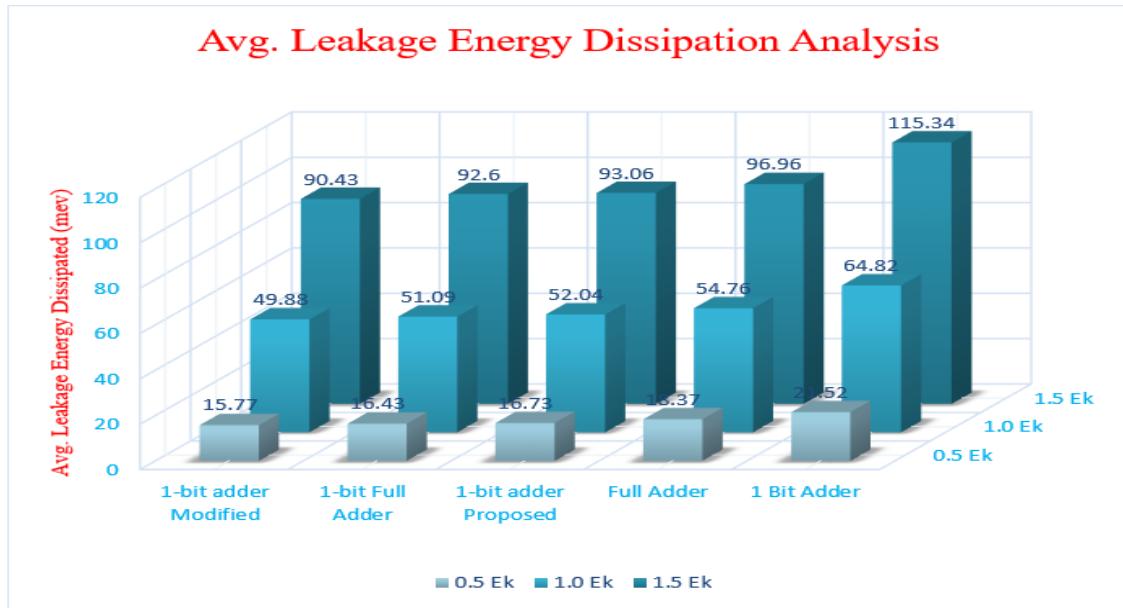


Figure 7.38: Avg. Leakage Energy dissipation (mev) for Adder designs at $T=1^0K$

Avg. Switching Energy Dissipation Analysis

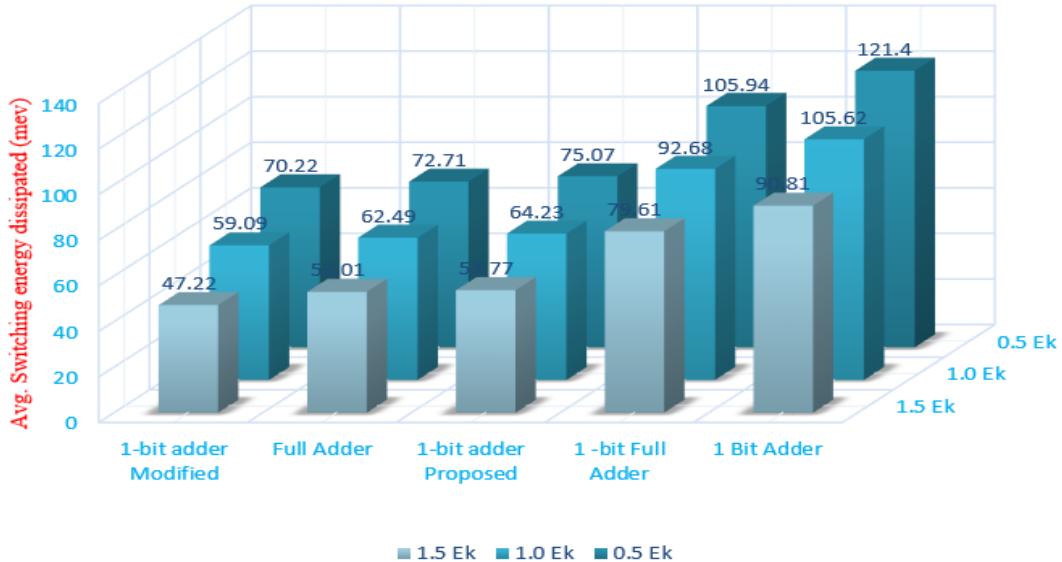


Figure 7.39: Avg. Switching Energy dissipation (mev) for Adder designs at $T=1^0 K$

Avg. Energy Dissipated in Circuit Analysis

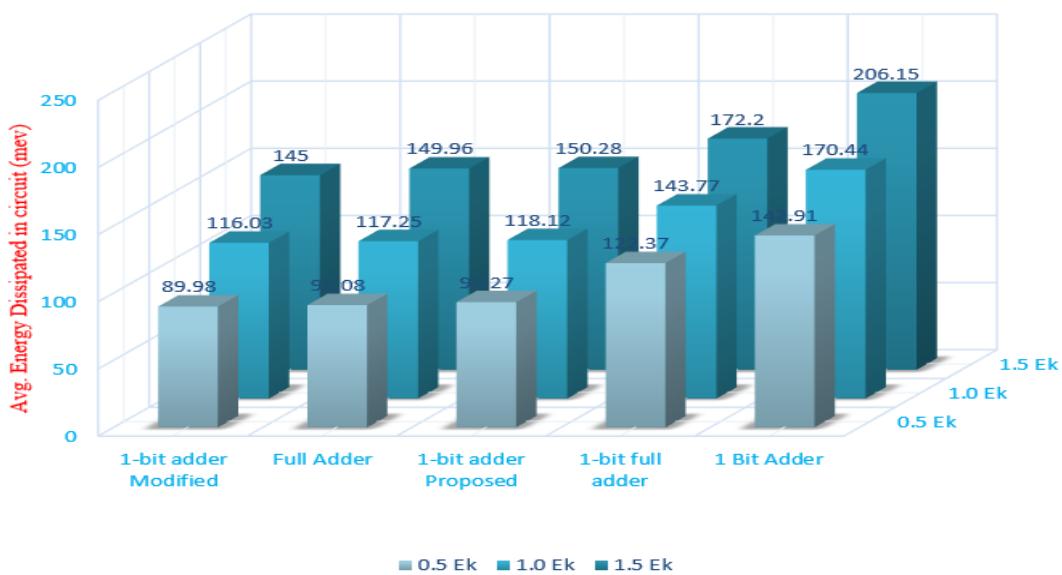


Figure 7.40: Avg. Energy dissipation in circuit (mev) for Adder designs at $T=1^0 K$

7.2.4 Reversible Gates

7.2.4.1 CNOT Gate

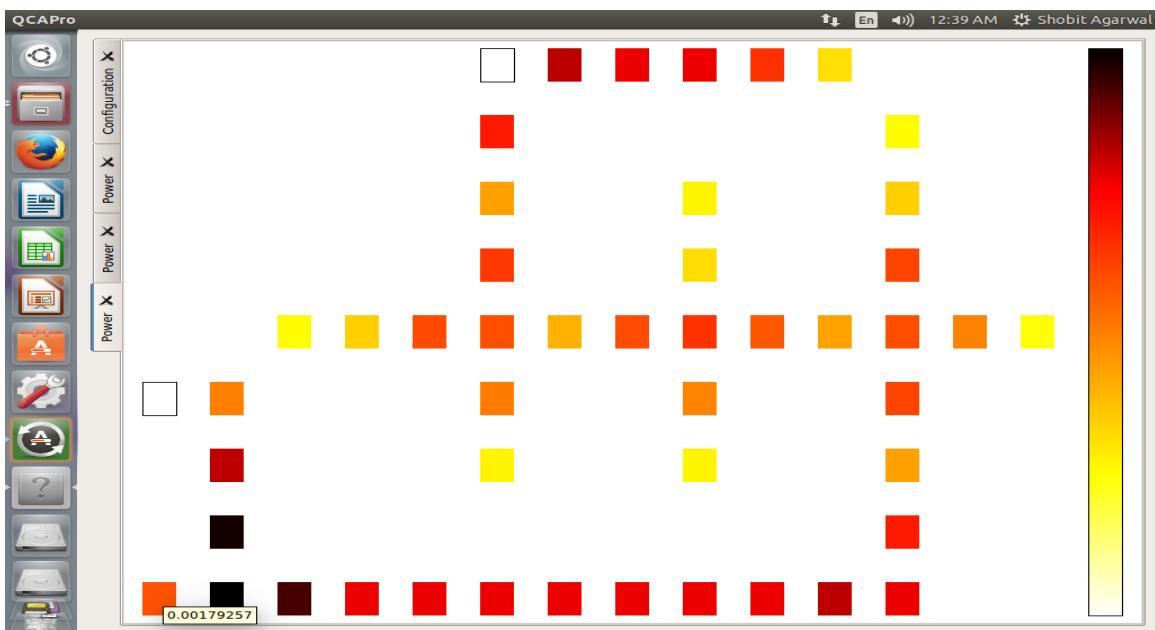


Figure 7.41: Power dissipation map for CNOT Gate $P(1, 0.5\text{Ek})$

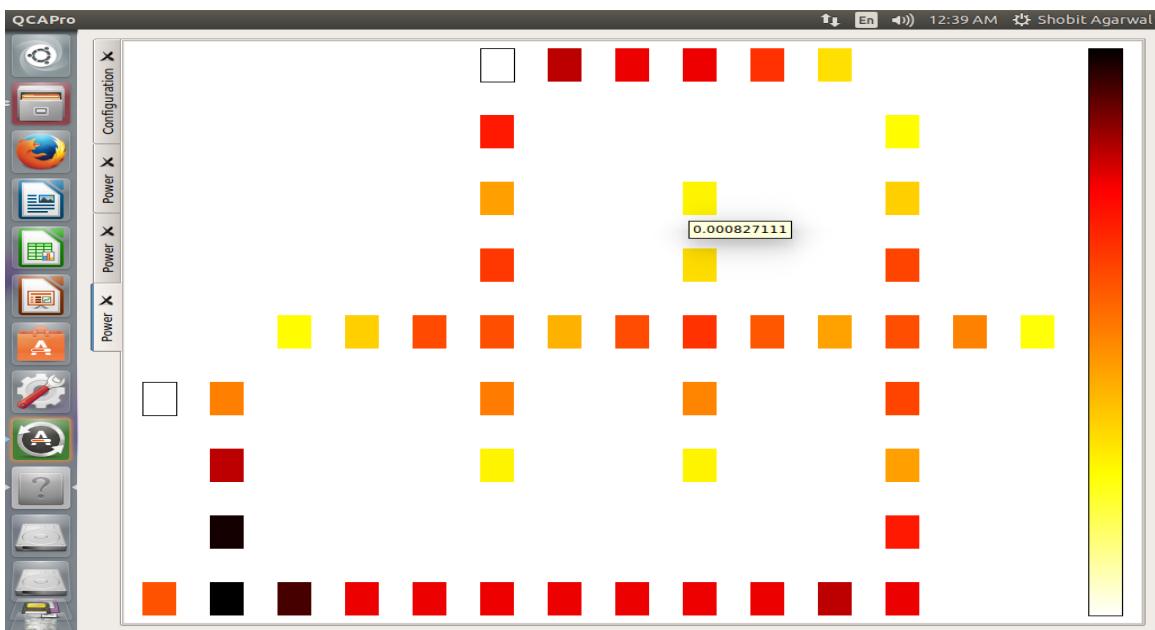


Figure 7.42: Power dissipation map for CNOT Gate $Q(1, 0.5\text{Ek})$

7.2.4.2 FEYNMAN Gate

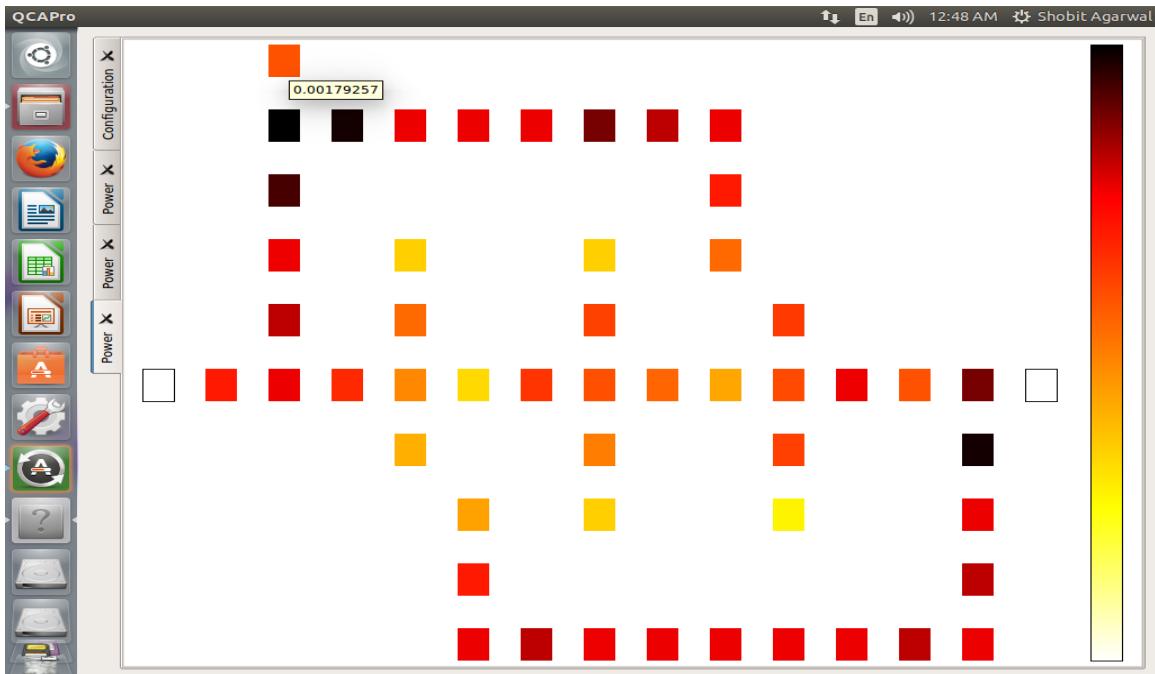


Figure 7.43: Power dissipation map for FEYNMAN Gate $P(1, 0.5Ek)$

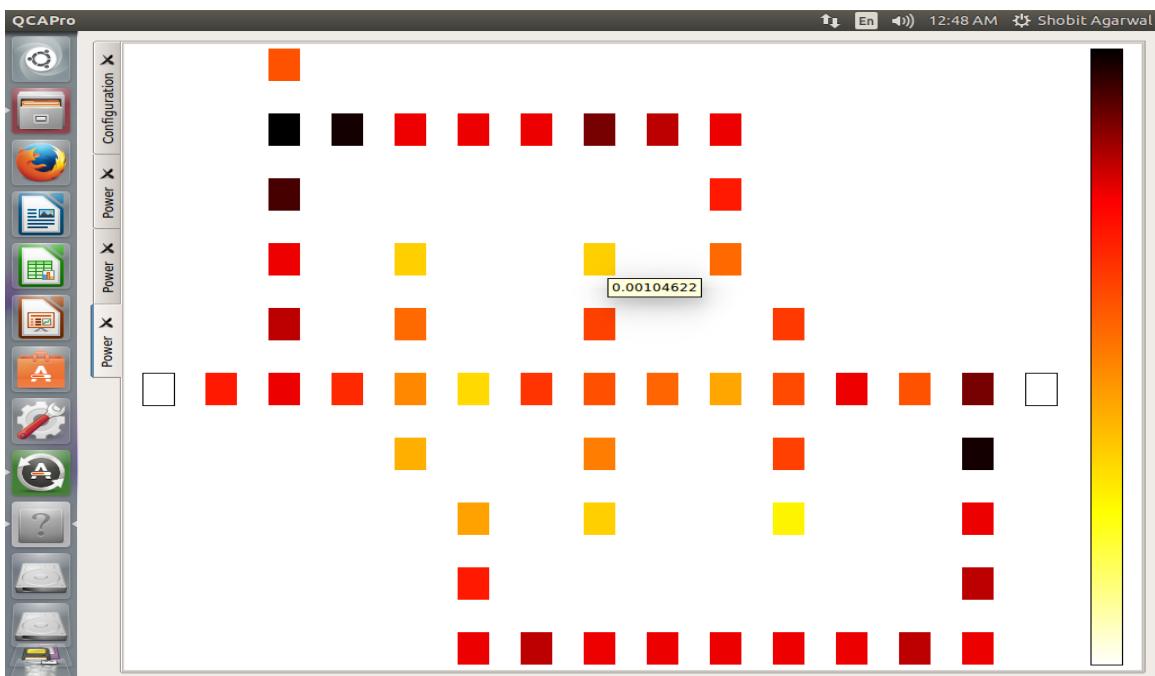


Figure 7.44: Power dissipation map for FEYNMAN Gate $Q(1, 0.5Ek)$

7.2.4.3 Double FEYNMAN Gate

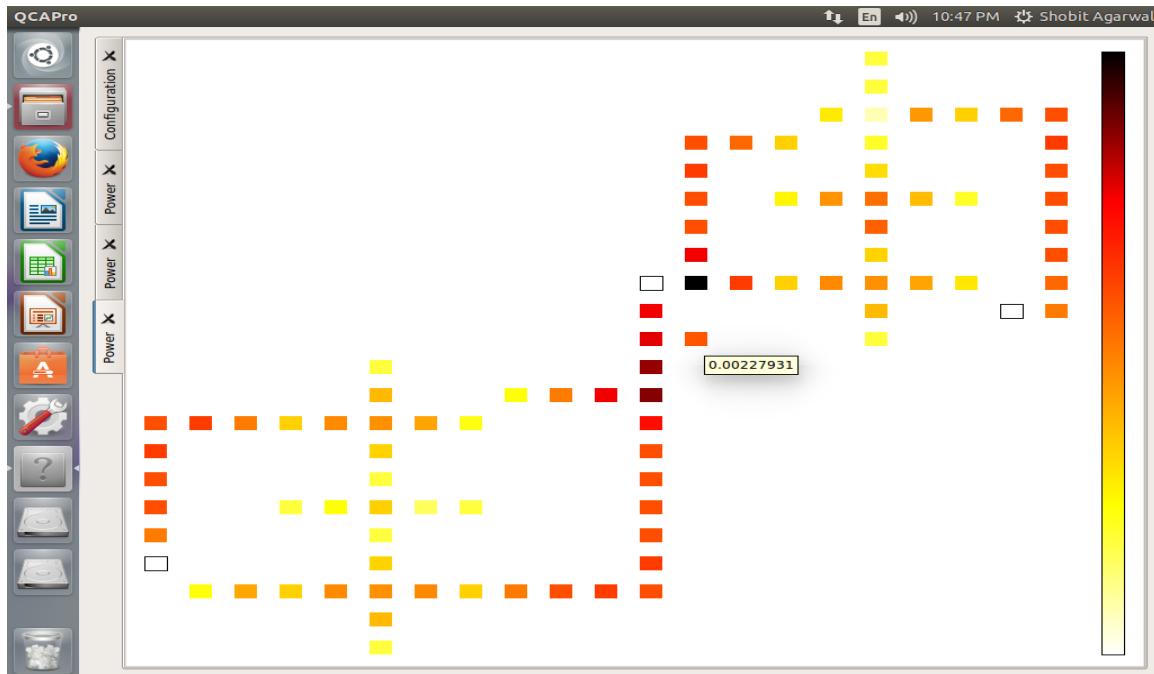


Figure 7.45: Power dissipation map for Double FEYNMAN Gate $P(1, 0.5Ek)$

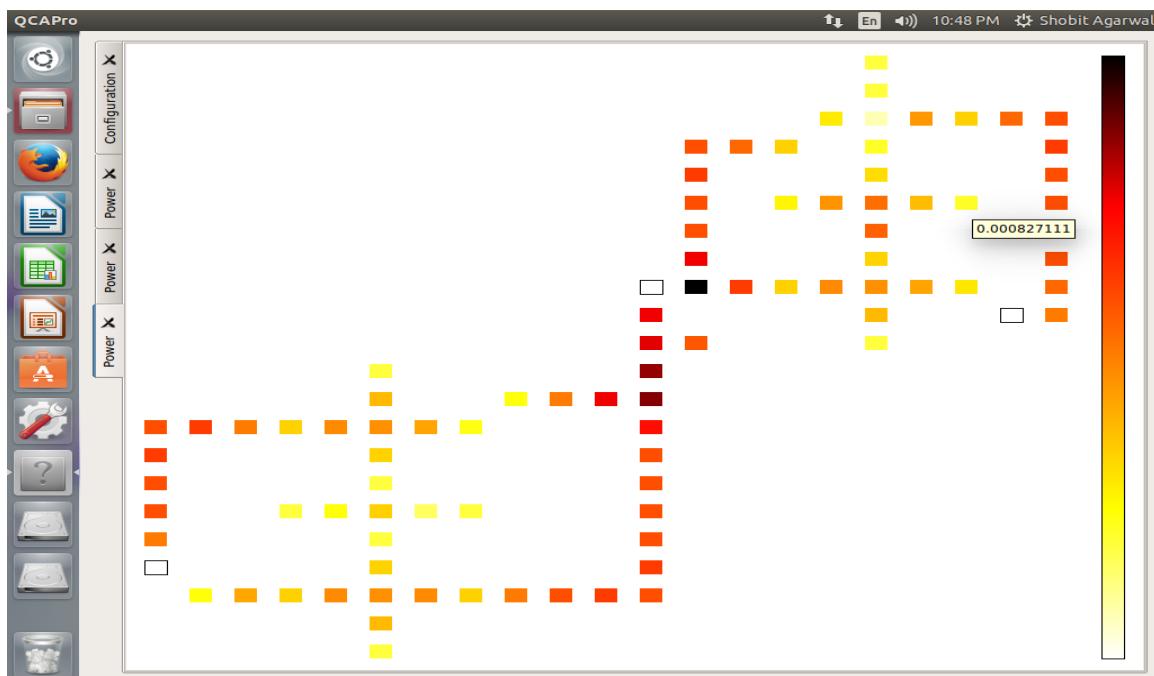


Figure 7.46: Power dissipation map for Double FEYNMAN Gate $Q(1, 0.5Ek)$

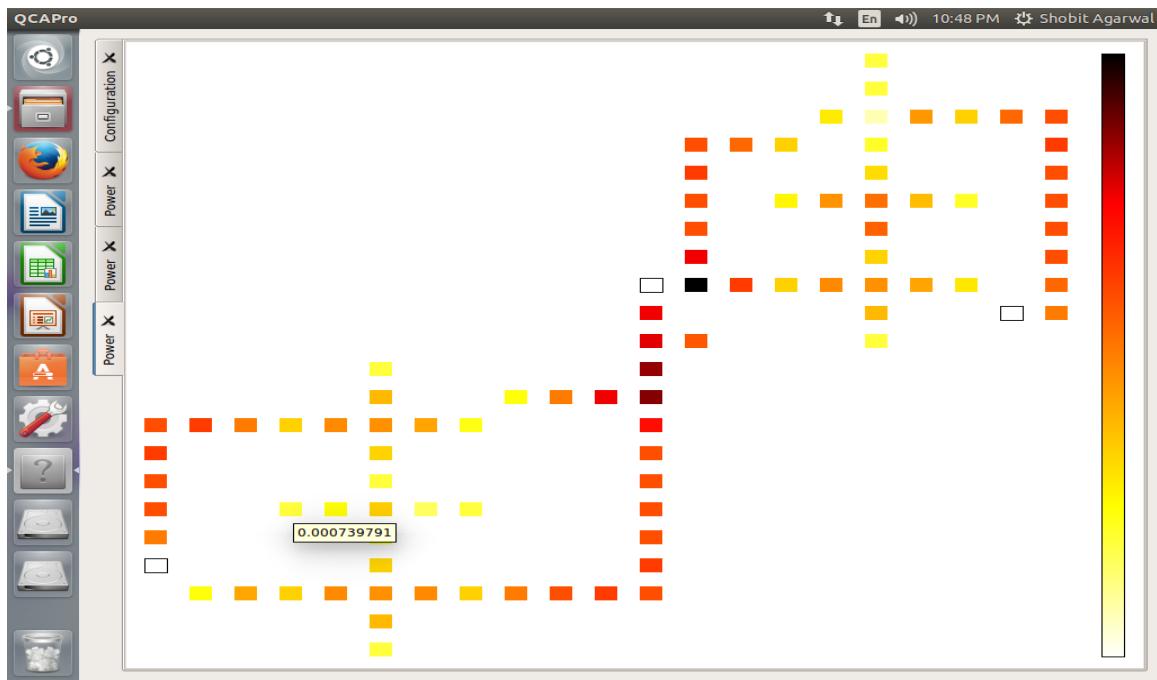


Figure 7.47: Power dissipation map for Double FEYNMAN Gate $R(1, 0.5Ek)$

7.2.4.4 TOFFOLI Gate



Figure 7.48: Power dissipation map for TOFFOLI Gate $P(1, 0.5Ek)$

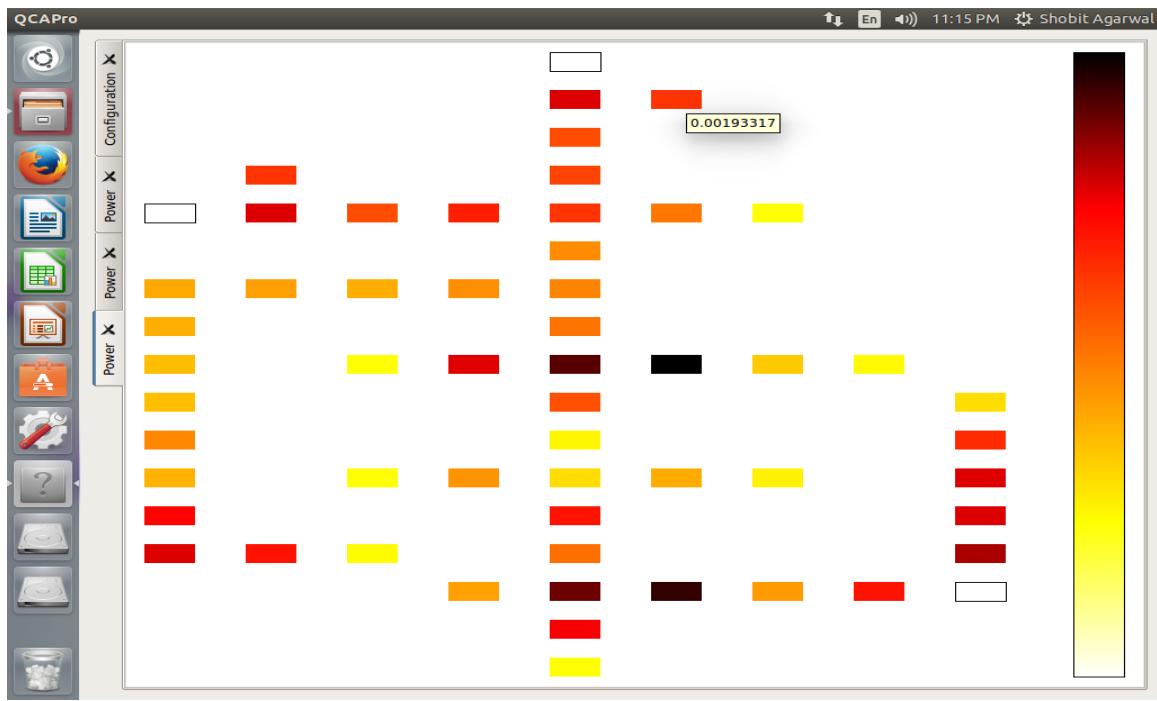


Figure 7.49: Power dissipation map for TOFFOLI Gate $Q(1, 0.5Ek)$



Figure 7.50: Power dissipation map for TOFFOLI Gate $R(1, 0.5Ek)$

7.2.4.5 FREDKIN Gate

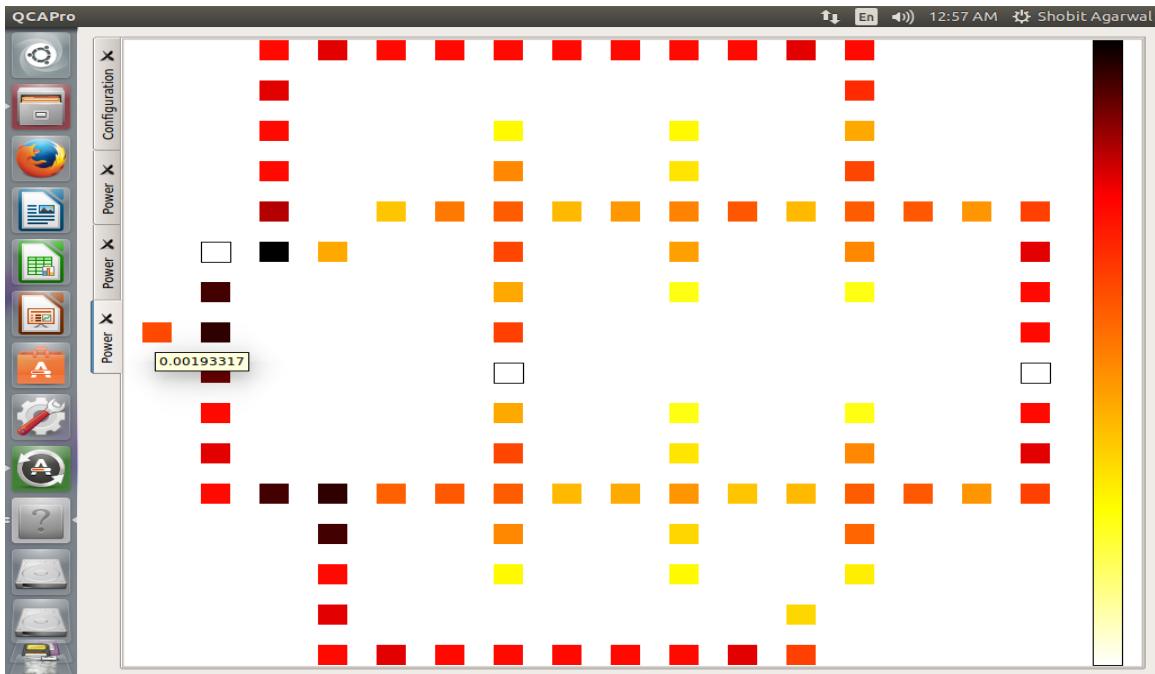


Figure 7.51: Power dissipation map for FREDKIN Gate $P(1, 0.5Ek)$

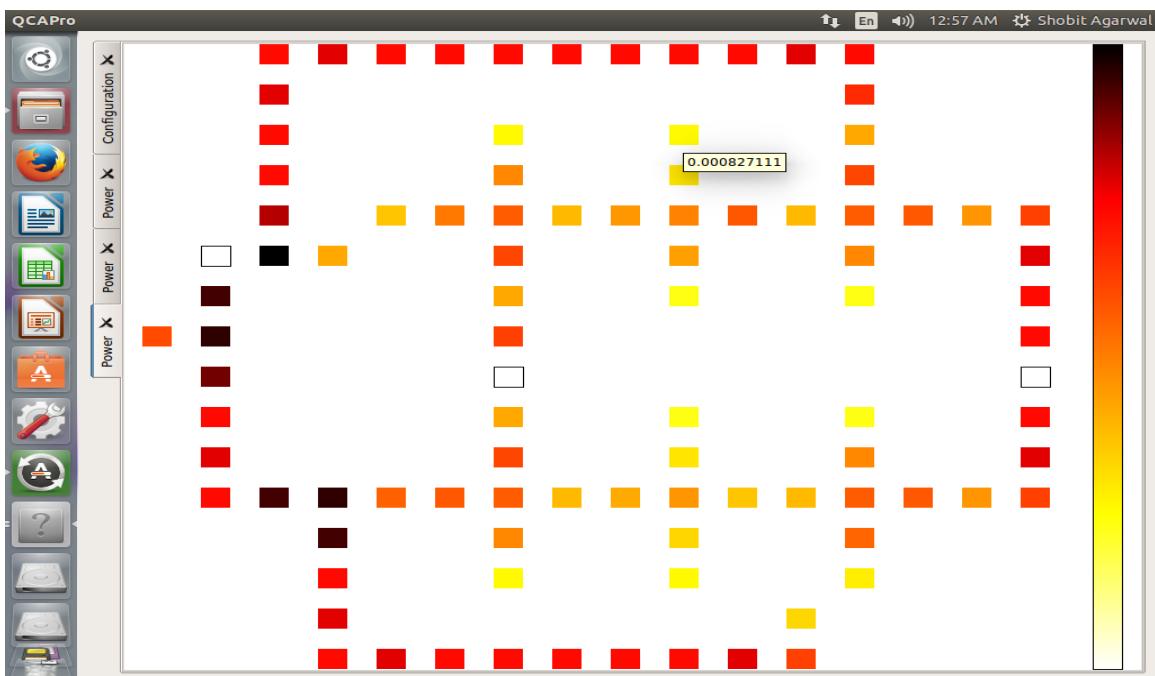


Figure 7.52: Power dissipation map for FREDKIN Gate $Q(1, 0.5Ek)$

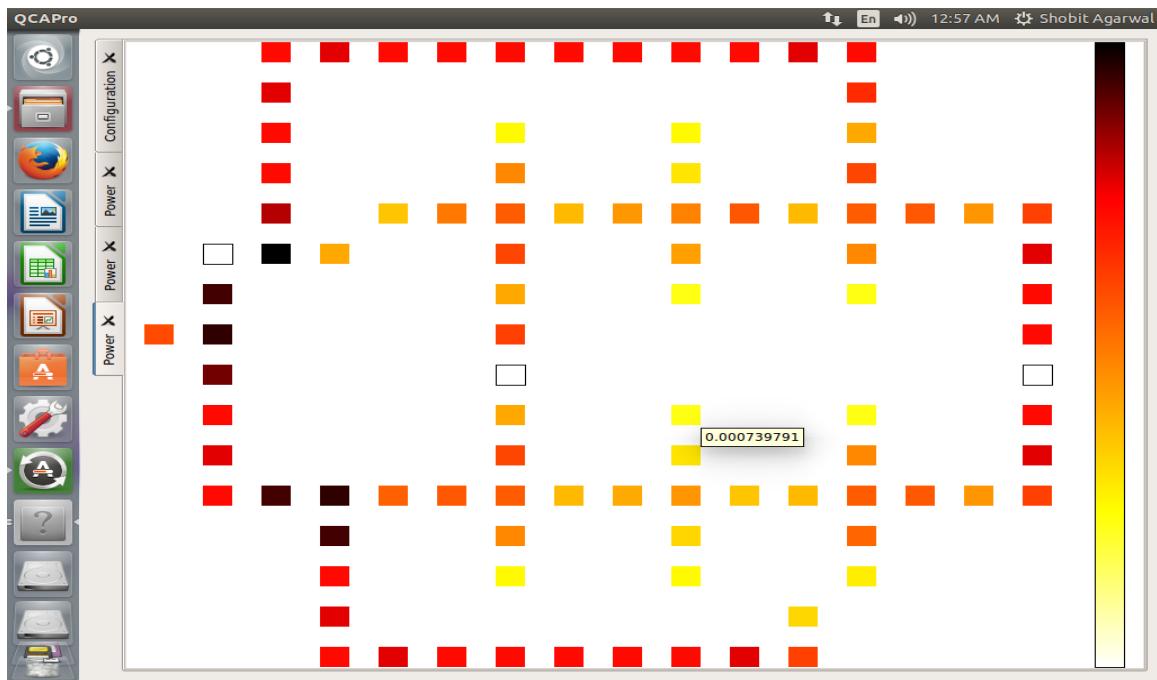


Figure 7.53: Power dissipation map for FREDKIN Gate $R(1, 0.5Ek)$

7.2.4.6 BJT Gate

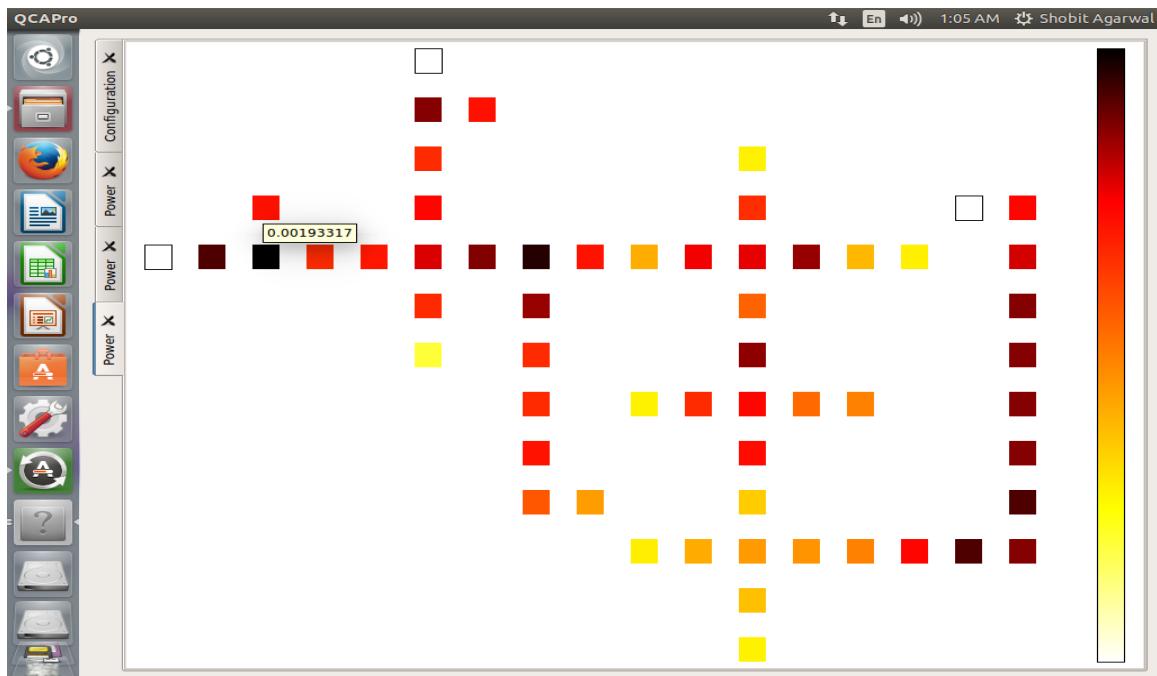


Figure 7.54: Power dissipation map for BJT Gate $P(1, 0.5Ek)$

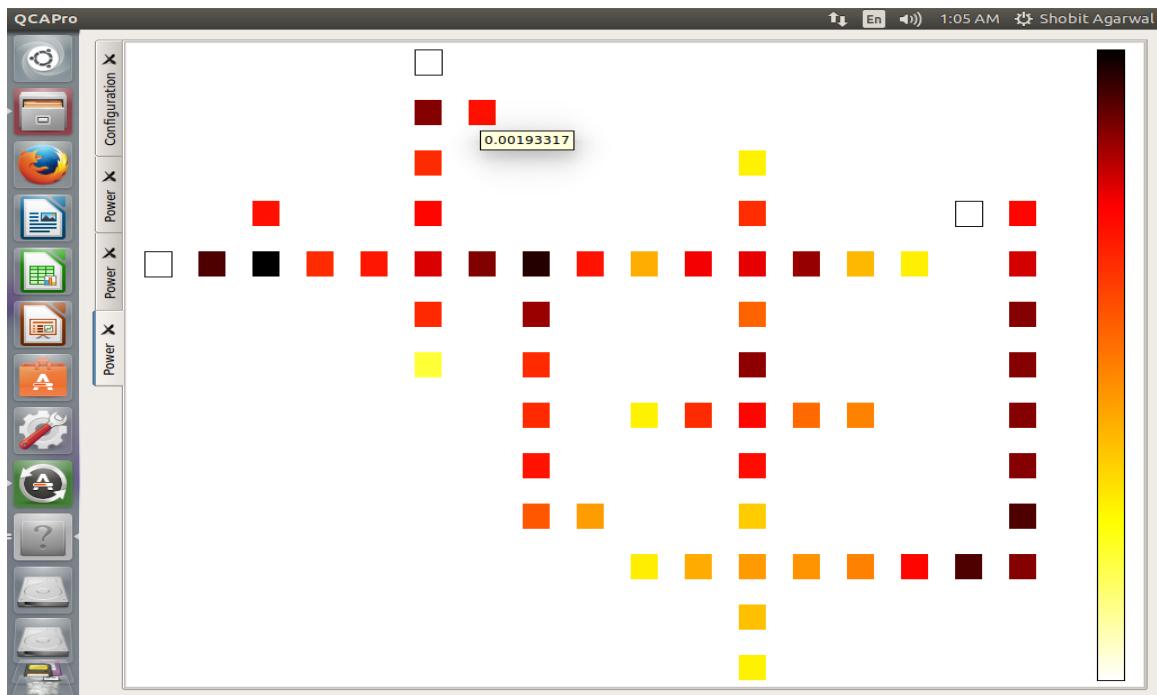


Figure 7.55: Power dissipation map for BJT Gate $Q(1, 0.5\text{Ek})$

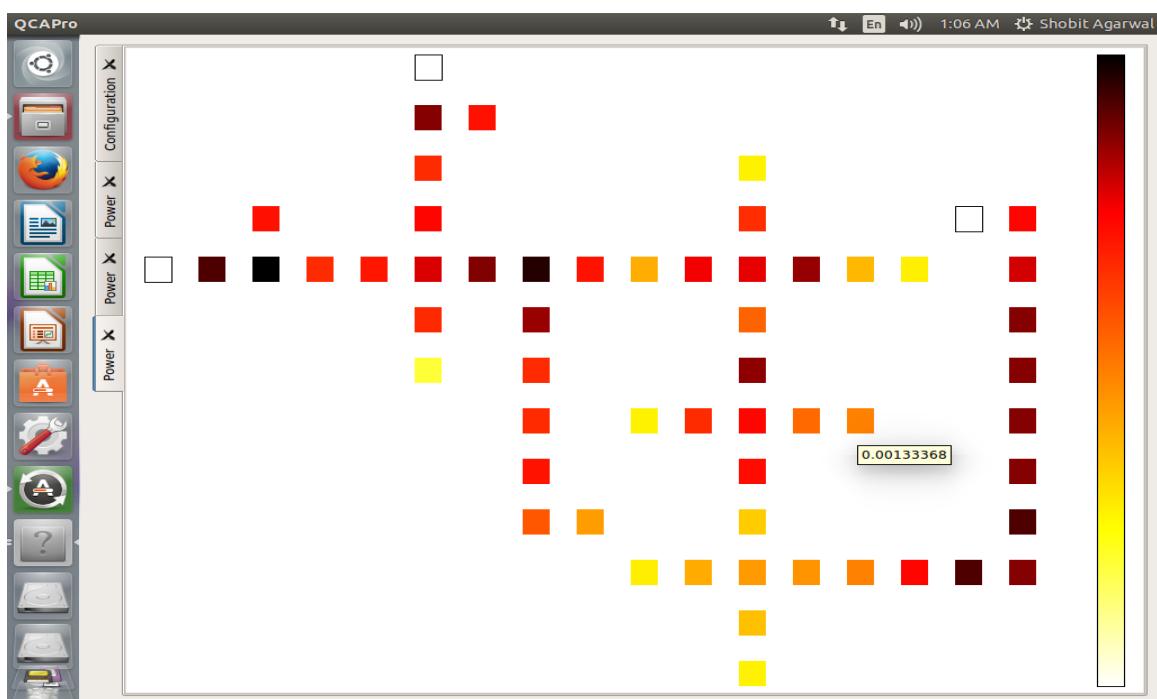


Figure 7.56: Power dissipation map for BJT Gate $R(1, 0.5\text{Ek})$

7.2.4.7 MCL Gate

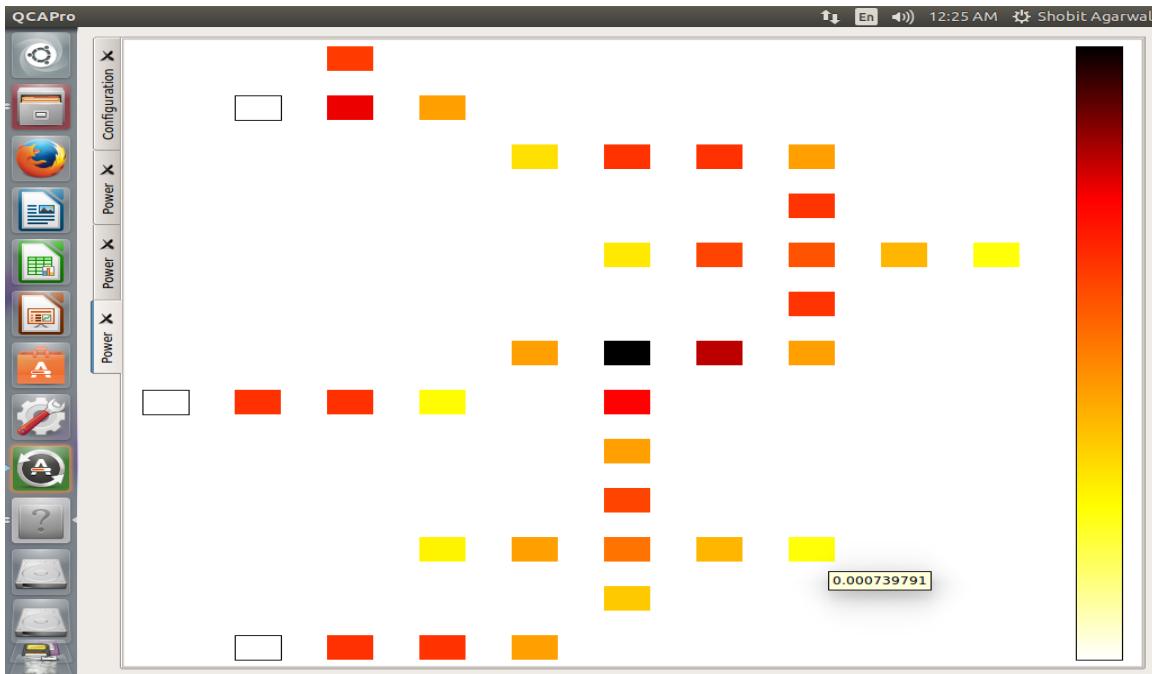


Figure 7.57: Power dissipation map for MCL Gate $P(1, 0.5Ek)$

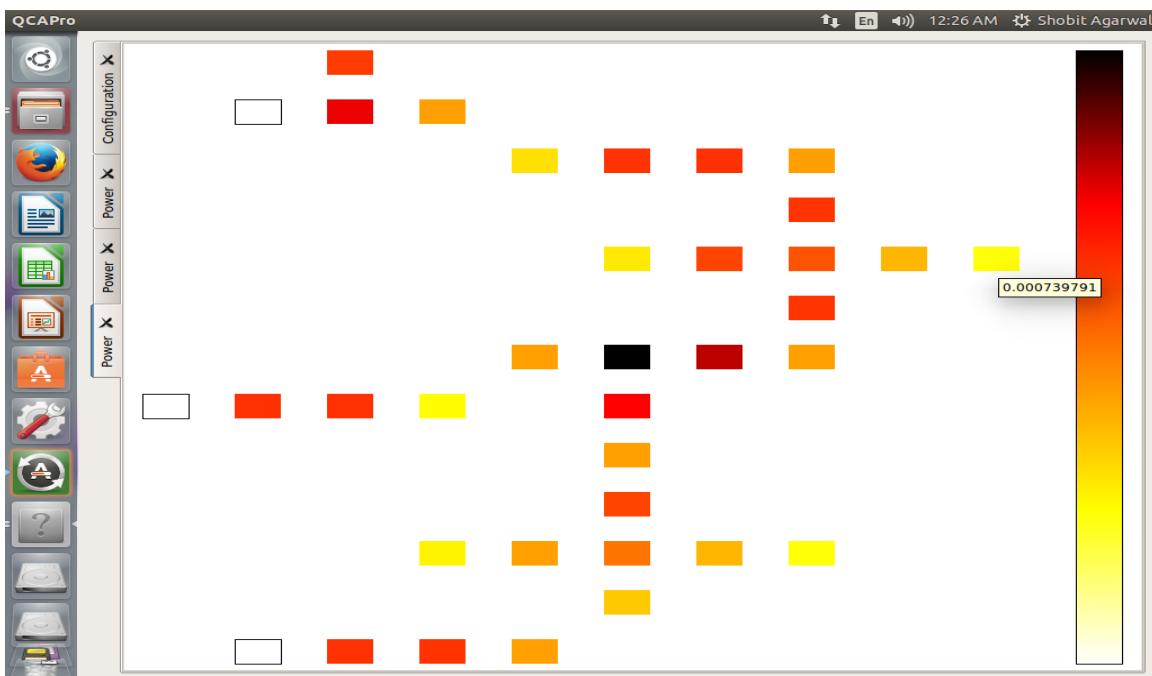


Figure 7.58: Power dissipation map for MCL Gate $Q(1, 0.5Ek)$

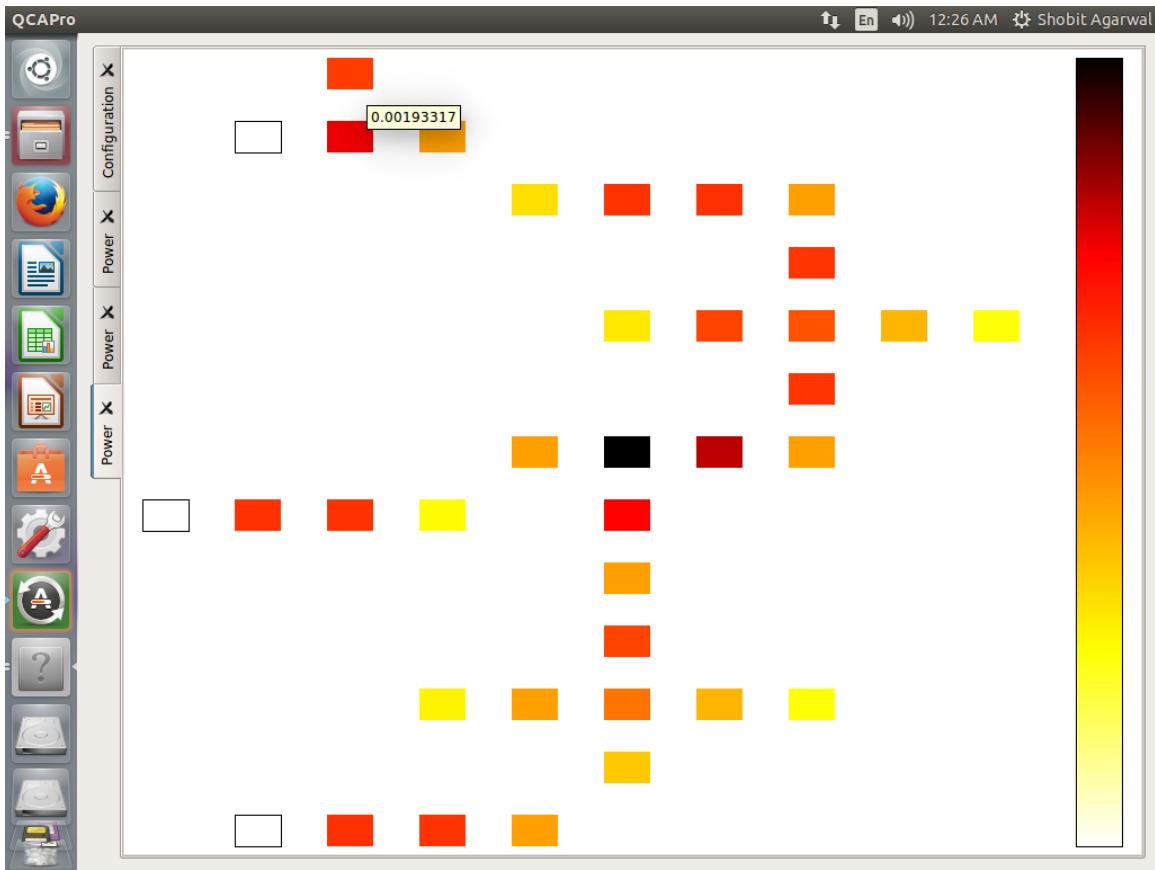


Figure 7.59: Power dissipation map for MCL Gate $R(1, 0.5E_k)$

7.2.4.8 Performance Analysis of Reversible Gates

	0.5 E_k	1.0 E_k	1.5 E_k
MCL	12.18	34.29	59.31
CNOT	16.10	46.55	81.91
FEYNMAN	15.04	46.89	85.12
BJN	17.50	52.57	93.92
TOFFOLI	19.16	53.77	93.98
Double FEYNMAN	29.66	86.76	153.94
FREDKIN	29.70	89.03	159.47

Table 7.14: Avg. Leakage Energy dissipation (mev) at $T=1^0K$

	1.5 E_k	1.0 E_k	0.5 E_k
MCL	28.37	34.07	40.67
CNOT	49.77	58.88	68.50
TOFFOLI	50.96	60.06	69.14
BJN	57.94	68.47	79.28
FEYNMAN	66.05	77.86	89.90
Double FEYNMAN	99.67	116.85	134.40
FREDKIN	109.08	128.40	148.03

Table 7.15: Avg. Switching Energy dissipation (mev) at $T=1^0K$

	0.5 E_k	1.0 E_k	1.5 E_k
MCL	52.85	68.36	87.68
CNOT	84.60	105.43	131.68
TOFFOLI	88.80	113.83	144.94
BJN	96.77	121.03	151.86
FEYNMAN	104.94	124.75	151.18
Double FEYNMAN	164.06	203.61	253.61
FREDKIN	177.73	217.43	268.55

Table 7.16: Avg. Energy dissipation in circuit (mev) at $T=1^0 K$

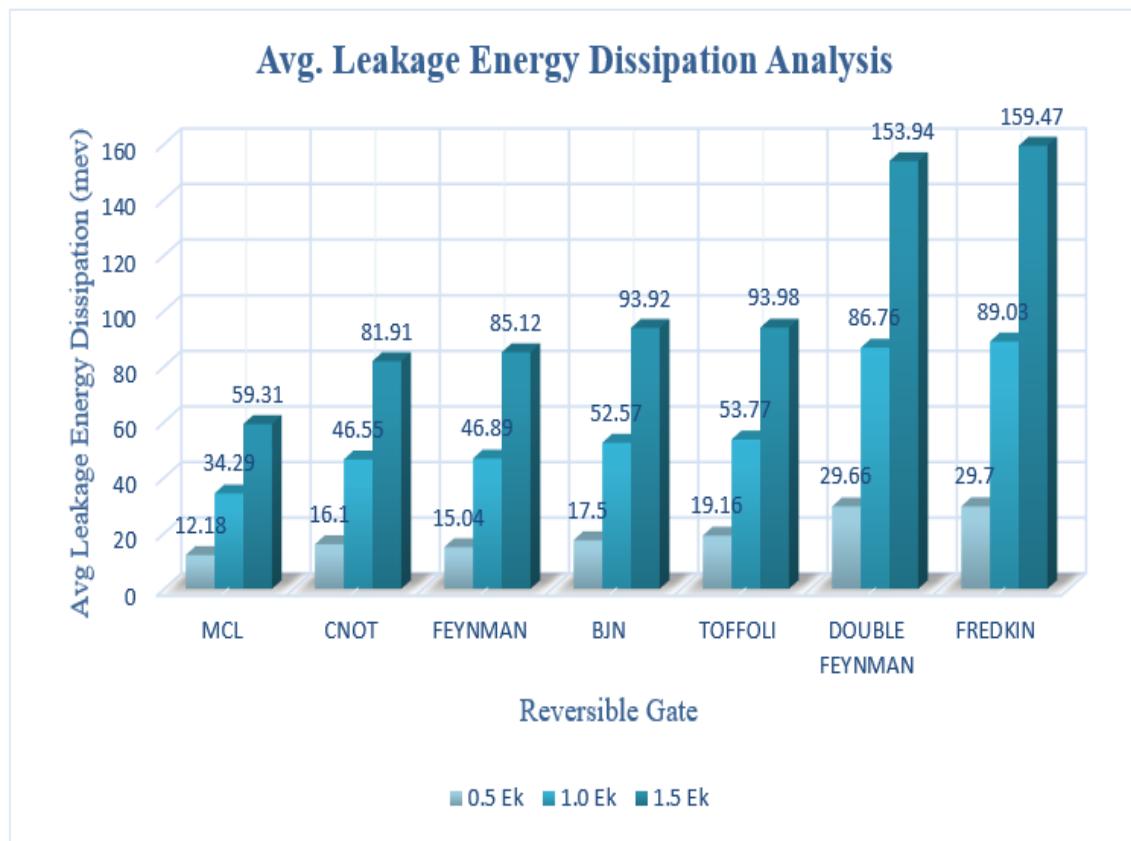


Figure 7.60: Avg. Leakage Energy dissipation (mev) for Reversible Gates at $T=1^0 K$

Average Switching energy Dissipation Analysis

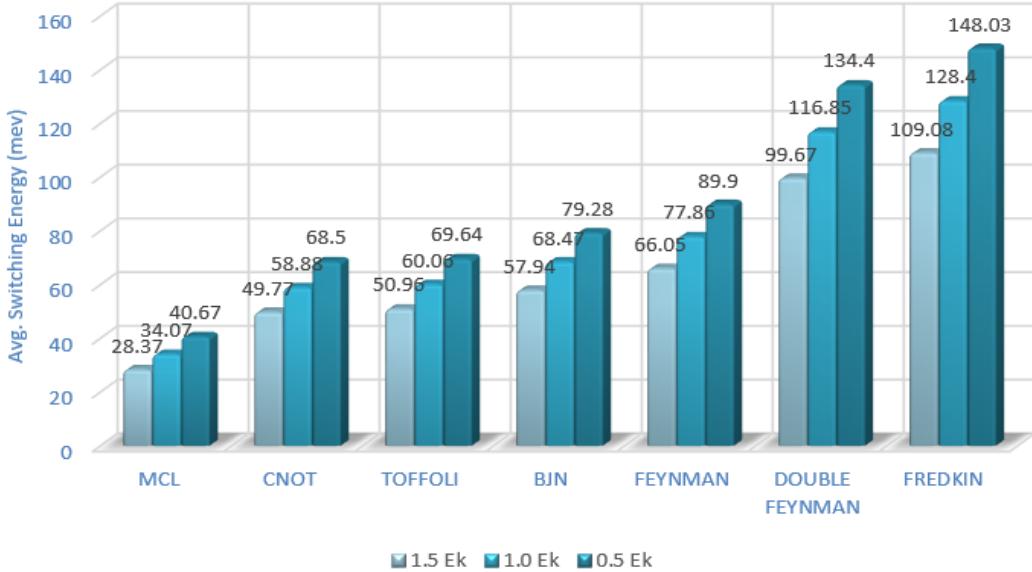


Figure 7.61: Avg. Switching Energy dissipation (mev) for Reversible Gates at $T=1^0K$

Avg. Energy Dissipation in circuit Analysis

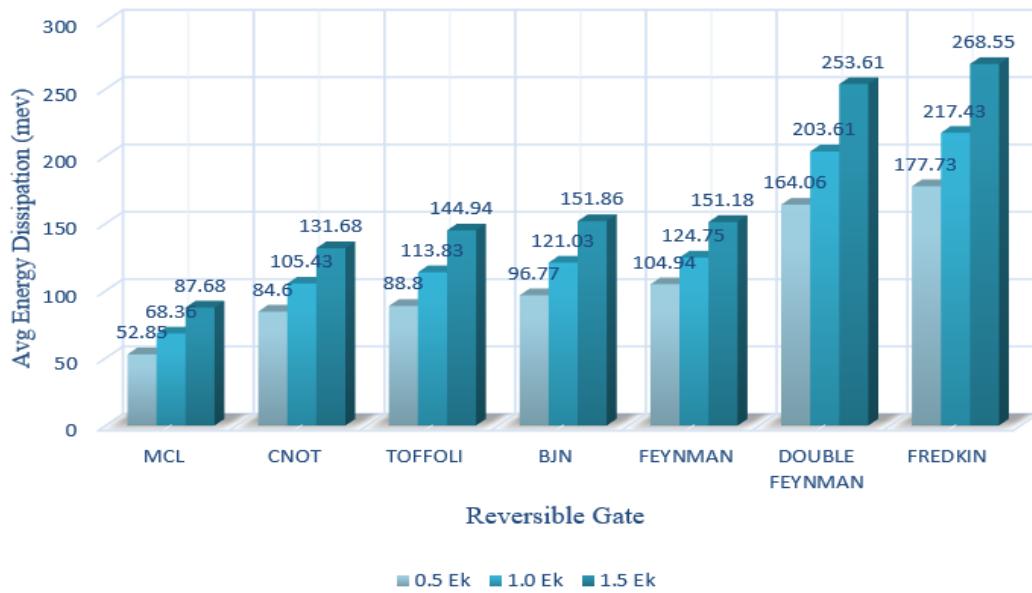


Figure 7.62: Avg. Energy dissipation in circuit (mev) for Reversible Gates at $T=1^0K$

Chapter 8

Conclusions and Future Work

First of all, by using QCADesigner tool 2.0.3, we have implemented and simulated various digital circuits viz. Logic Gates, Multiplexers, Adders, Subtractor, Multipliers, Random Access memory. Later we focussed on some adder designs and proposed some new designs for 1-bit and 2-bit adder. According to QCADesigner tool version 2.0.3, these adders uses a total of 87 & 218 cells respectively for 1-bit and 2-bit adder. These designs uses a cell of $18nm$ with diameter of $5nm$ and spacing of $2nm$ between cells. According to QCA tool 1-bit adder, 1-bit adder modified and 2 bit adders uses area as of following:

$$1 \text{ bit adder: } 384nm \times 381nm = 146304 \text{ nm}^2 = 0.15 \mu\text{m}^2$$

$$1 \text{ bit adder modified: } 344nm \times 300nm = 103200 \text{ nm}^2 = 0.10 \mu\text{m}^2$$

$$2 \text{ bit adder: } 656.09nm \times 581.00nm = 381188.21 \text{ nm}^2 = 0.38 \mu\text{m}^2$$

After designing of adders, we focussed on designing of adder/subtractor circuit and proposed a new 8 bit adder subtractor circuit here. Since power dissipation is a major factor in VLSI technology, therefore we moved forward to power dissipation analysis carried out by QCA Pro tool. We tried to analyse as many circuits as we could have done.

8.1 Scope of future work

In future, we can focus on designing of multipliers which can be used along with adders to implement an Arithmetic and Logical Unit (ALU). Which in turn can be used to design memory circuits. ALU is an important part of any computational device as all arithmetic and logical operations are performed inside ALU.

Bibliography

- [1] C. S. Lent and P. D. Tougaw, “A device architecture for computing with quantum dots,” *Proceedings of the IEEE*, vol. 85, no. 4, pp. 541–557, 1997.
- [2] W.-T. J. Chan, A. B. Kahng, S. Nath, and I. Yamamoto, “The itrs mpu and soc system drivers: Calibration and implications for design-based equivalent scaling in the roadmap,” in *2014 IEEE 32nd International Conference on Computer Design (ICCD)*. IEEE, 2014, pp. 153–160.
- [3] S. Bhanja and S. Srivastava, “A bayesian computing model for qcas,” in *NSTI Nanotechnology Conference*, 2005.
- [4] K. Walus, T. J. Dysart, G. A. Jullien, and R. A. Budiman, “Qcadesigner: A rapid design and simulation tool for quantum-dot cellular automata,” *IEEE transactions on Nanotechnology*, vol. 3, no. 1, pp. 26–31, 2004.
- [5] S. Srivastava, A. Asthana, S. Bhanja, and S. Sarkar, “Qcapro—an error-power estimation tool for qca circuit design,” in *2011 IEEE International Symposium of Circuits and Systems (ISCAS)*. IEEE, 2011, pp. 2377–2380.
- [6] S. Sheikhfaal, S. Angizi, S. Sarmadi, M. H. Moaiyeri, and S. Sayedsalehi, “Designing efficient qca logical circuits with power dissipation analysis,” *Microelectronics Journal*, vol. 46, no. 6, pp. 462–471, 2015.
- [7] 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, no. 2, pp. 259–271, 2014.
- [8] A. M. Chabi, S. Sayedsalehi, S. Angizi, and K. Navi, “Efficient qca exclusive-or and multiplexer circuits based on a nanoelectronic-compatible designing approach,” *International Scholarly Research Notices*, vol. 2014, 2014.
- [9] J. Timler and C. S. Lent, “Maxwell’s demon and quantum-dot cellular automata,” *Journal of Applied Physics*, vol. 94, no. 2, pp. 1050–1060, 2003.

- [10] S. Angizi, S. Sarmadi, S. Sayedsalehi, and K. Navi, “Design and evaluation of new majority gate-based ram cell in quantum-dot cellular automata,” *Microelectronics Journal*, vol. 46, no. 1, pp. 43–51, 2015.
- [11] S. Hashemi, R. Farazkish, and K. Navi, “New quantum dot cellular automata cell arrangements,” *Journal of Computational and Theoretical Nanoscience*, vol. 10, no. 4, pp. 798–809, 2013.
- [12] S. Sayedsalehi, M. H. Moaiyeri, and K. Navi, “Novel efficient adder circuits for quantum-dot cellular automata,” *Journal of Computational and Theoretical Nanoscience*, vol. 8, no. 9, pp. 1769–1775, 2011.
- [13] M. R. Beigh, M. Mustafa, and F. Ahmad, “Performance evaluation of efficient xor structures in quantum-dot cellular automata (qca),” 2013.
- [14] B. Sen, M. Goswami, S. Mazumdar, and B. K. Sikdar, “Towards modular design of reliable quantum-dot cellular automata logic circuit using multiplexers,” *Computers & Electrical Engineering*, vol. 45, pp. 42–54, 2015.
- [15] I. Hanninen and J. Takala, “Robust adders based on quantum-dot cellular automata,” in *2007 IEEE International Conf. on Application-specific Systems, Architectures and Processors (ASAP)*. IEEE, 2007, pp. 391–396.
- [16] M. R. Azghadi, O. Kavehie, and K. Navi, “A novel design for quantum-dot cellular automata cells and full adders,” *arXiv preprint arXiv:1204.2048*, 2012.
- [17] S. Hashemi and K. Navi, “A novel robust qca full-adder,” *Procedia Materials Science*, vol. 11, pp. 376–380, 2015.
- [18] F. Ahmad, G. M. Bhat, H. Khademolhosseini, S. Azimi, S. Angizi, and K. Navi, “Towards single layer quantum-dot cellular automata adders based on explicit interaction of cells,” *Journal of Computational Science*, vol. 16, pp. 8–15, 2016.
- [19] A. M. Chabi, A. Roohi, R. F. DeMara, S. Angizi, K. Navi, and H. Khademolhosseini, “Cost-efficient qca reversible combinational circuits based on a new reversible gate,” in *Computer Architecture and Digital Systems (CADS), 2015 18th CSI International Symposium on*. IEEE, 2015, pp. 1–6.
- [20] F. Ahmad, G. M. Bhat, and P. Z. Ahmad, “Novel adder circuits based on quantum-dot cellular automata (qca),” *Circuits and Systems*, vol. 2014, 2014.
- [21] K. Hennessy and C. S. Lent, “Clocking of molecular quantum-dot cellular automata,” *Journal of Vacuum Science & Technology B*, vol. 19, no. 5, pp. 1752–1755, 2001.
- [22] S. Hashemi and K. Navi, “New robust qca d flip flop and memory structures,” *Microelectronics Journal*, vol. 43, no. 12, pp. 929–940, 2012.

- [23] M. Kianpour, R. Sabbaghi-Nadooshan, and K. Navi, “A novel design of 8-bit adder/subtractor by quantum-dot cellular automata,” *Journal of Computer and System Sciences*, vol. 80, no. 7, pp. 1404–1414, 2014.
- [24] M. Abdullah-Al-Shafi, M. Shifatul, and A. Newaz, “A review on reversible logic gates and its qca implementation,” *International Journal of Computer Applications*, vol. 128, no. 2, pp. 27–34, 2015.
- [25] B. Sen, A. Rajoria, and B. K. Sikdar, “Design of efficient full adder in quantum-dot cellular automata,” *The Scientific World Journal*, vol. 2013, 2013.
- [26] A. O. Orlov, R. Kummamuru, R. Ramasubramaniam, C. S. Lent, G. H. Bernstein, and G. L. Snider, “Clocked quantum-dot cellular automata shift register,” *Surface Science*, vol. 532, pp. 1193–1198, 2003.
- [27] R. Kummamuru, A. Orlov, J. Timler, R. Ramasubramaniam, C. Lent, G. Bernstein, and G. Snider, “A quantum-dot cellular automata shift register,” in *Device Research Conf*, 2001, pp. 25–27.
- [28] M. A. Dehkordi, A. S. Shamsabadi, B. S. Ghahfarokhi, and A. Vafaei, “Novel ram cell designs based on inherent capabilities of quantum-dot cellular automata,” *Microelectronics Journal*, vol. 42, no. 5, pp. 701–708, 2011.
- [29] A. S. Shamsabadi, B. S. Ghahfarokhi, K. Zamanifar, and N. Movahedinia, “Applying inherent capabilities of quantum-dot cellular automata to design: D flip-flop case study,” *Journal of Systems Architecture*, vol. 55, no. 3, pp. 180–187, 2009.
- [30] S. Srivastava, S. Sarkar, and S. Bhanja, “Estimation of upper bound of power dissipation in qca circuits,” *IEEE Transactions on Nanotechnology*, vol. 8, no. 1, pp. 116–127, 2009.
- [31] S. Bhanja and S. Sarkar, “Probabilistic modeling of qca circuits using bayesian networks,” *IEEE Transactions on Nanotechnology*, vol. 5, no. 6, pp. 657–670, 2006.
- [32] R. Akeela and M. D. Wagh, “A five-input majority gate in quantum-dot cellular automata,” in *NSTI Nanotech*, vol. 2, 2011, pp. 978–981.
- [33] M. A. Tehrani, F. Safaei, M. H. Moaiyeri, and K. Navi, “Design and implementation of multistage interconnection networks using quantum-dot cellular automata,” *Microelectronics Journal*, vol. 42, no. 6, pp. 913–922, 2011.
- [34] W. Wang, K. Walus, and G. A. Jullien, “Quantum-dot cellular automata adders,” in *Nanotechnology, 2003. IEEE-NANO 2003. 2003 Third IEEE Conference on*, vol. 1. IEEE, 2003, pp. 461–464.
- [35] R. Zhang, K. Walus, W. Wang, and G. A. Jullien, “Performance comparison of quantum-dot cellular automata adders,” in *2005 IEEE International Symposium on Circuits and Systems*. IEEE, 2005, pp. 2522–2526.

- [36] V. Pudi and K. Sridharan, “Low complexity design of ripple carry and brent–kung adders in qca,” *IEEE Transactions on Nanotechnology*, vol. 11, no. 1, pp. 105–119, 2012.
- [37] V. Vankamamidi, M. Ottavi, and F. Lombardi, “A serial memory by quantum-dot cellular automata (qca),” *IEEE Transactions on Computers*, vol. 57, no. 5, pp. 606–618, 2008.
- [38] A. N. Bahar, S. Waheed, M. A. Uddin, and M. A. Habib, “Double feynman gate (f2g) in quantum-dot cellular automata (qca),” *Int. J. Comput. Sci. Eng.*, vol. 2, pp. 351–355, 2013.
- [39] B. R. Kanth, B. M. Krishna, M. Sridhar, and V. S. Swaroop, “A distinguish between reversible and conventional logic gates,” 2012.