

**M. Tech. dissertation on**  
***Exploration, Design and Analysis of reversible circuits using  
quantum computation.***

**IS SUBMITTED IN PARTIAL FULFILLMENT OF  
THE MASTER OF TECHNOLOGY IN EMBEDDED SYSTEMS**



**Submitted By**  
**GARIMA SHARMA**  
**2017PEB5409**

**Internal Supervisor**  
**Rakesh Bairathi**  
**Associate Professor,**  
**E.C.E Dept.**  
**MNIT, Jaipur**

**External Supervisor**  
**Mr. Biswajit Chikkabelavangala**  
**FC Integration & Clocking Manager,**  
**INTEL Technology India Pvt. Ltd.**  
**Bengaluru**

**Department of Electronics and Communication Engineering**  
**Malaviya National Institute of Technology, Jaipur, India**  
**June, 2019**

*© Malaviya National Institute of Technology, Jaipur All rights reserved*

## CERTIFICATE



**Department of Electronics and Communication  
Malaviya National Institute of Technology, Jaipur Rajasthan – 302017**

This is to certify that the dissertation report entitled “*Exploration, Design and Analysis of reversible circuits using quantum computation*” submitted by Garima Sharma (2017PEB5409), in the partial fulfilment of the Degree Master of Technology in Embedded Systems of Malaviya National Institute of Technology, is a work completed by her under our supervision, and approved for submission during academic session 2018-2019.

**Rakesh Bairathi  
Associate Professor,  
E.C.E Dept.  
MNIT, Jaipur**

**Mr. Biswajit Chikkabelavangala  
FC Integration & Clocking Manager,  
INTEL Technology India Pvt. Ltd.  
Bengaluru**

## DECLARATION

This is to certify that the dissertation report entitled “*Exploration, Design and Analysis of reversible circuits using quantum computation*” is being submitted by me in partial fulfillment of degree of Master of Technology in Embedded Systems during 2017-19. This work is carried out by me under the supervision of Rakesh Bairathi ,Associate Prof., Deptt of ECE, MNIT, Jaipur and Mr. Biswajit Chikkabelavangala, FC Integration & Clocking Manager, INTEL Technology India Pvt. Ltd, Bengaluru . I am fully responsible for the material used in this report in case if any discrepancies arises. The report (fully/partly) is not submitted for the award of any other degree. Wherever I have consulted the published work(in any form) of others, it is clearly attributed. Where I have quoted from the work of others, the source is always given. I have acknowledged all the relevant (to to best of my knowledge) sources in the report. With the exception of above , this report is entirely my own work.

Garima Sharma  
(2017PEB5409)

## ACKNOWLEDGMENTS

I take this opportunity to express my deep sense of gratitude and respect towards my Supervisor (Guide), Rakesh Bairathi , Associate Professor, Department of Electronics & Communication Engineering, Malaviya National Institute of Technology. I am very much indebted to him for the generosity, expertise and guidance, I have received from him while working on this project and throughout my studies. Without his support and timely guidance, the completion of my project would have seemed a far-fetched dream. In this respect, I find myself fortunate to have him as my Project Guide. He has guided me not only with the subject matter, but also taught me the proper approach, style and techniques of working.

I would like to thank my mentor Mr. Biswajit Chikkabelavangala, FC Integration & Clocking Manager,INTEL Technology India Pvt. Ltd, Bengaluru, for his co-operation and support in this project work during my internship. I am also thankful to my manager Mr. Narendra Nimmagadda for his constant support throughout my internship in INTEL. I am grateful to him in guiding me in the successful completion of this project work.

I would like to express my gratitude and sincere thanks to our Head of Department (HoD), Dr. D. Boolchandani, Malaviya National Institute of Technology (MNIT) Jaipur for allowing me to undertake this dissertation work and for his guidelines during the review process.

I take this opportunity to express my regards and obligation to my father and other family members whose support and encouragement, has always been a guiding force for me to work harder.

Lastly, I am thankful to all those who have supported me directly or indirectly during the dissertation work.

Garima Sharma  
(2017PEB5409)

## ABSTRACT

Reversible logic has attracted much attention due to the ability to reduce the power wastage in recent years, which is the main requirement for low power design. At present, the conventional Boolean algebra based calculations with the use of silicon-based semiconductor technology are irreversible in nature. This is due to the mismatch between the number of inputs and outputs.

The reverse logic circuit maps the unique input for output and ensures one-to-one mapping and acts as a strong base for emerging applications such as low-power design, quantum computation, optical computing and nanotechnology. In the computing paradigm, the building of an Arithmetic Logic Unit (ALU) computer is one of intensive building blocks that operate arithmetic and logical operations.

This work understands and nurtures the need for reversible ALUs for future revolutionary computing technologies. Arithmetic logic unit (ALU) is one of the most important components of any system and is used for various devices such as calculator, cell phone and computer and so on. But ALUs that are built using traditional digital or non-reverse logic gates, they consume more power. So to reduce the power consumption, reverse logic technology helps in a long time.

This research provides the implementation of 2-bit, 4-bit and 6-bit ALU based on reversible logic to reduce power consumption during its arithmetic and logical operations. Two different techniques are used for the implementation of a multi bit ALU and analysis is done in the context of garbage output, ANCILLA input and quantum cost.

## LIST OF FIGURES

Fig 1 Classical vs Quantum bit	8
Fig 2 Qubit on a bloch sphere.	9
Fig 3 Quantum entanglement	11
Fig 4 Illustration of classical vs quantum states	13
Fig 5 Josephson junction model	14
Fig 6 A simple cooper pair box diagram	16
Fig 7 Diagram of a SQUID device.	16
Fig 8 Current biased josephson junction	17
Fig 9 The latest INTEL's 49 qubit quantum processor	19
Fig 10 NOT gate	25
Fig 11 CNOT gate	25
Fig 12 Toffoli gate	25
Fig 13 Quantum equivalent	27
Fig 14 Reversible gate and its Quantum equivalent	28
Fig 15 Reversible gate and its Quantum equivalent	28
Fig 16 Reversible gate and its Quantum equivalent	29
Fig 17 Reversible gate and its Quantum equivalent	29
Fig 18 Reversible gate and its Quantum equivalent	29
Fig 19 Conventional ALU	30
Fig 20 Two Conventional ALU	31
Fig 21 Reversible AND gate	32
Fig 22 Reversible XOR gate	33
Fig 23 Fig 23 Reversible ADDER	33
Fig.24 Reversible two bit ALU	33
Fig 25 Quantum representation 2 bit ALU using Method I	34
Fig 26 Quantum representation 2 bit ALU using Method II	35
Fig 27 AND/NAND quantum structure	37
Fig 28 XOR/XNOR quantum structure	39
Fig 29 XOR/XNOR quantum structure in the main ALU	39
Fig 30 XOR/XNOR quantum structure	39
Fig 31 XOR/XNOR quantum structure in the main ALU	40
Fig 32 ADDER/SUBTRACTOR quantum structure	40
Fig 33 ADDER/SUBTRACTOR quantum structure in the main ALU	41
Fig 34 An abstract version of the 4 bit ALU	42
Fig 35 Four bit ALU representation with exact pins	43
Fig 36 Quantum representation of 4 bit ALU, using Method II	44
Fig 37 An abstract version of the 6 bit ALU	46
Fig 38 Six bit ALU representation with exact pins	47
Fig 39 Quantum representation of 6 bit ALU, using Method II	48

## LIST OF TABLES:

Table 1 – CNOT Gate	21
Table 2 – Toffoli Gate	22
Table 3 – Fredkin Gate	23
Table 4 – Fenyman Gate	23
Table 5 – Peres Gate	24
Table 6 – Truth table for synthesis reference	26
Table 7 – Bidirectional synthesis algorithm	27
Table 8 – Operation table for a 2 bit ALU	32
Table 9 - Operation table for a 2 bit ALU	33
Table 10 - Operation table for a 2 bit ALU	36
Table 11 - Operation table for a 4 bit ALU	37
Table 12 - Operation table for a 6 bit ALU	45
Table 13 - Results for 2 Bit ALU usign Method I and Method II	51
Table 14 - Enhancements in ALU functionality using Method II	51

# CONTENTS

<b>Certificate</b>	<b>ii</b>
<b>Declaration</b>	<b>iii</b>
<b>Acknowledgments</b>	<b>iv</b>
<b>Abstract</b>	<b>v</b>
<b>List of Figures</b>	<b>vi</b>
<b>List of Tables</b>	<b>vii</b>
1. CHAPTER 1 – Introduction	1
1.1 Motivation	1
1.2 Introduction	2
2. CHAPTER 2 – Literature survey	3
2.1 Literature survey	3
3. CHAPTER 3 – Into the realm of reversible logic gates	5
3.1 Reversible logic gates	5
3.2 Quantum computing for reversible logic	6
3.3 Qubit	7
3.4 Properties of qubit	11
3.5 Physical interpretation of qubit	14
3.6 Josephson junction and Qubits	14
3.7 Differences between classical and reversible computing	20
4. CHAPTER 4 – Simulation and Synthesis of reversible logic circuits	21
4.1 Family of basic reversible gates	21
4.2 Synthesis of reversible gates	24
4.3 Reversible gates with their quantum equivalent	28
5. CHAPTER 5 – Proposed Work and Results	30
5.1 Reversible ALU circuit design and synthesis	30
5.1.1 Method I	30
5.1.2 Method II	36
5.1.3 Circuit parameters calculation	49
5.1.4 Results and Analysis	51
6. CHAPTER 6 -Conclusion and Future Work	52
6.1 Conclusion	52
6.2 Future Work	53
6.3 References	54



# **CHAPTER 1 - Introduction**

## **1.1 MOTIVATION**

We live in a world where the dependency on digital devices is growing at a rate faster than ever before. These digital devices are made to process more and more information in the least possible time frame. The amount of information to be processed by these devices is ever growing, in order to process this ever increasing volume of information more and more components are being fabricated on a the least possible chip area following the Moore's law. Over the past few decades, the size of the components has been reduced in order to increase the density of these components on integrated circuits. However, this pattern cannot continue forever because the current technology is approaching the physical limits of computation. Also limitations of traditional computing on the heat dissipation facet is also becoming a bigger factor of consideration for digital devices.

Reversible computing offers a solution to this potential deadlock of further development in traditional computing.

This is where reversible computing comes into picture. Due to the ability to reduce power wastage, reversible logic has attracted much attention in recent years, which is the main requirement in low power design. It requires a study of reliable reversible computing calculation models that demonstrates both the forward and backward fixes. While reversible computing initially gained interest through its ability to reduce the consumption of computing machinery, as a result of physical science, now known as Landauer's [1] theory, many other applications in computer science have been proposed and have been done, from syntax details to models based testing, debugging and even robotics. It is needed to set up reversible computing as a solid on a foundation, because for conventional computing existing technologies need both critical adaptation and development of new ones.

In this dissertation, we investigate reversible computing from a logical point of view, broadly construed.

## 1.2 INTRODUCTION

According to Charles Bennet[2] ,who has played a major role in clarifying the interconnection between physics and information, especially in the realm of quantum calculation, but also in cellular automata and reversible computing, theoretically zero power dissipation can be achieved only when the circuit is logically reversible. A very important factor in modern VLSI design is the result of irreversible hardware computing resulting in energy dissipation due to information loss. According to R. Landauer for an immutable logic computation, the information loss for one bit creates  $KT\ln 2$  joule of heat energy, where K is the constant of Boltzmann, T is the temperature on which to be calculated. At room temperature, the dissipating heat is around  $2.9 \times 10^{-21}$  J.This figure of energy loss defined by Landauer is given importance because it is very likely that the growth of heat generation will be very noticeable in future and will impose an inevitable limitation on further scaling of digital devices in accordance to Moore's law.

Reversible computing is one of the most innovative and efficient attempt at designing low power dissipation circuits which reduces the abovementioned power dissipation by eliminating information loss.Because of rapid switching in the internal signals in modern VLSI system, there is high power dissipation. Logically, zero power dissipation in a logic circuit is only possible when the circuit comprises of reversible logic gates[3]. Reversible logic has become one of the most promising areas for study in low power design domain. Over the past years, its application has been found in many research areas and many technologies such as low power CMOS, nano computing and optical computing. Reversible logic gates are dominantly known to be compatible with future computing technologies which are capable of dissipating virtually zero heat.

For this reason, in the recent years, reversible computing has introduced itself as a promising area of research and emerging technologies. Other reasons for research into reversible systems include connections to quantum computing[4], and applications in cryptography , nano-computing technologies and digital processing .

It is widely believed that the next generation computers will be quantum computers. Quantum computations are reversible, and the circuits in quantum computing work on reversible functions [5]. Therefore, research on reversible logic is inevitable from various points of view.Presently, the logic gates used in traditional computation are not reversible.Also the relationship between the input and the output of a traditional logic gate is many to one. For example, an AND gate has two or more inputs and one output. Hence the relationship between the input and output is many to one.However, the relationship between the input and the output of a reversible gate is bijective.Over the past few years, several reversible logic gates and synthesis methods have been proposed.However, analysing the logic gates and related synthesis approaches to achieve the most optimized reversible circuits is still an active area of research.

## CHAPTER 2 – Literature Survey

### 2.1 LITERATURE SURVEY

1) R. Landauer, —Irreversibility and Heat Generation in the Computational Process, IBM Journal of Research and Development, vol. 5, pp. 183-191, 1961. R. Landauer's showed, the amount of energy (heat) dissipated for every irreversible bit operation is given by  $KT \ln 2$ , where  $K$  is the Boltzmann's constant ( $1.3807 \times 10^{-23} \text{ JK}^{-1}$ ) and  $T$  is the operating temperature. At room temperature (300 K),  $KT \ln 2$  is approximately  $2.8 \times 10^{-21} \text{ J}$ , which is small but not negligible. He also showed that only the logically irreversible steps in a computation carry an unavoidable energy penalty. If we could compute entirely with reversible operations, there would be no lower limit on energy consumption.

2) C.H. Bennett, "Notes on the History of Reversible Computation", IBM Journal of Research and Development, vol. 32, pp. 16-23, 1998.[3] Bennett showed that  $kT \ln 2$  energy dissipation would not occur, if a computation is carried out in a reversible way, since the amount of energy dissipated in a system bears a direct relationship to the number of bits erased during computation.

3) Lihui Ni, Zhijin Guan, and Wenying Zhu, —A General Method of Constructing the Reversible Full-Adder, Third International Symposium on Intelligent Information Technology and Security Informatics, pp.109-113, 2010. Lihui Ni, Zhijin Guan, and Wenying Zhu described general approach to construct the Reversible full adder and can be extended to a variety of Reversible full adders with only two Reversible gates.

4) Bruce, J.W., M.A. Thornton, L. shivakuamaraiah, P.S. kokate and X. Li, —Efficient adder circuits based on a conservative reversible logic gate, IEEE computer society Annual symposium on VLSI, Pittsburgh, Pennsylvania, and pp: 83-88, 2000. Bruce, J.W., M.A. Thornton, L. shivakuamaraiah, P.S. kokate and X. Li, used only Fredkin gates to construct full adder with gates cost equal to 4, 3 garbage outputs and 2 constant input.

5) Anamika, Rockey Bhardwaj , - “Reversible Logic Gates and its Performances”, Proceedings of the Second International Conference on Inventive Systems and Control (ICISC 2018) IEEE Xplore – they constructed a full adder circuit using only peres gates which takes only 1 ancilla input bit and two garbage output bit.

6) Shaveta Thakral, Dipali Bansal, S.K. Chakarvarti, “Review of Truth Table Based Reversible Logic Synthesis Methods”, 2015 International Conference on Soft Computing Techniques and Implementations- (ICSCTI) Department of ECE, FET, MRIU, Faridabad, India, Oct 8-10, 2015 – There are many synthesis techniques

available for reversible logic synthesis like BDD based, Truth Table based etc. This paper proposed a truth table based synthesis technique for construction of a reversible circuit by making use of only toffoli gates in the quantum implementation. The input to the synthesis technique is a truth table representation of the desired reversible function and after going through the synthesis algorithm, a quantum network consisting of only toffoli gates is constructed.

7) Arindam Banerjee, Debesh Kumar Das, "A New ALU Architecture Design using Reversible Logic", 2016 Sixth International Symposium on Embedded Computing and System Design (ISED). This work proposed a novel approach to synthesize a reversible ALU with the use of modular quantum functional blocks to be compiled by multiplexing using select input bits. This approach proves to be very good in terms of quantum cost, ancilla input bits and garbage output bits.

## CHAPTER 3 - Into the realm of reversible logic gates

### 3.1 REVERSIBLE LOGIC GATES

A gate is said to be reversible, if only its every input combination gives out a unique output combination. There should be one relation between the input and output of the reverse gate. Thus the input for a reversible gate can easily be determined by the given output combination and vice versa. A reversible gate is considered reversible when it has equal number of input and output. Output vectors for an N bit reversible gate are the permutations of 0 to  $2^N - 1$ . A reversible gate is balanced, that is, the number of useful output bits is half the number of inputs. Without a constant on its inputs, a circuits realized with reversible gates are only the balanced functions.

Non balanced functions can only be realized when there are garbage outputs in the reversible gate.

However, along with the fact of reversible circuits being of so much importance for future low power devices, it has a set of major limitations which are of very high concern such as

- 1) Fan-outs of reversible gates cannot be used.
- 2) A feedback from a gate output to a gate input cannot be made into use.

For any gate to be reversible , the following features are needed in it:

- 1) The number of inputs and outputs have to be the same.
- 2) Feedback or a loop isnt allowed in reversible logics.
- 3) Fan-out is defined as the maximum number of inputs that the output of a logic gate can feed. In reversible logics, Fan-out is not allowed .

From the reversible circuit design point of view, the complexity and performance of a circuit depends on some of the parameters as below:

**Constant Inputs:** The inputs's number which are to be maintained at a constant 0 or 1 value so as to obtain the required function.

**Garbage Output:** The number of unused outputs which transforms the irreversible circuit into a reversible circuit.

**Quantum Cost:** Number of Primitive reversible logic gates required to realize the circuit.

In order to achieve the most optimized reversible circuit, the following points are to be taken care of:

- 1) Minimum basic reversible gates should be used in number for the implementation of a reversible circuit.
- 2) Minimizing the garbage outputs produced.
- 3) Using a minimum number of constant inputs.

The number of gates of a circuit is considered an important complexity measure in traditional computing. In addition to this measure, the quantum cost and number of garbage lines are two additional factors to take into account in determining the complexity of a reversible circuit.

## **WHAT IS QUANTUM:**

Quantum theory is a revolutionary advancement in physics and chemistry that emerged in the beginning of the twentieth century. It is an elegant mathematical theory that is capable of explaining behavior in contrast to subatomic particles, especially the phenomenon of entanglement [6]. In the second half of the twentieth century, it was discovered that quantum theory not only applies to atoms and molecules but also applies to bits and logic operations in a computer. This realization is bringing a revolution in the field of science and technology in information processing, which makes it possible to make unknown types of computing and communication in the information age.

## **3.2 QUANTUM COMPUTING FOR REVERSIBLE LOGIC:**

Quantum computing technique is naturally reversible and is one of the most promising techniques for future computing systems. In addition to reversibility, it has some very strong properties which is a big advantage for the future computation systems. These properties are:

- 1) Quantum superposition
- 2) Quantum entanglement

The above two properties are explained in detail in the further sections of the dissertation.

These properties allow problems to be resolved more efficiently than current classical computing. For example, while a classical algorithm requires  $N$  steps to find a structured database, the same quantum algorithm proposed by Grover[17] requires  $\sqrt{N}$  steps only where  $N$  is number of elements in the searched unstructured database. Quantum circuit is reversible only when it performs the calculation in Hilbert's space.

## **WHAT IS HILBERT SPACE:**

Hilbert's space is a linear vector space. It follows the inner product property. Let us take two vectors  $\Phi_1$  and  $\Phi_2$  defined in Hilbert's space, describing the inner product operation property as follows:

## $\langle \Phi_1, \Phi_2 \rangle \in \text{Complex number}$

A vector in Hilbert's space represents the position of a physical system in quantum mechanics.

The word "Hilbert space" is often used to describe an infinite dimensions inner product space with a property that it is either full or closed. But nowadays, the word is often a misnomer and is used in a way that involves finite-dimensional spaces, which automatically satisfies the completeness condition.

The Dirac notation is used, in which the vectors in a space are denoted by  $|v\rangle$ , called a ket, where  $v$  is a symbol that identifies a vector. The inner product for the vectors  $|v\rangle$  with  $|w\rangle$ , in Dirac Notation, is written as  $\langle v|w\rangle$ . This is almost same to an ordinary dot product  $v \cdot w$  except that you take a complex vector conjugate on the left, so think about  $\overline{v}^* \cdot \overline{w}$

Hilbert's space are complete in nature i.e there are no gaps or no irrational number between any two numbers we are considering in the hilbert's space.

### 3.3 QUBITS:

The simplest and a very interesting space of hilbert's space is the two dimensional Hilbert's space. The space of a 2-dimensional Hilbert defines any vector in this space as a linear combination of two vectors, which forms the foundation for space.

A qubit describes the minimum state of a quantum system, which is described in the 2 dimensional Hilbert's space. A bit is denoted by the sign  $|x\rangle$ , where  $x$  is a quantum bit or a qubit. Every qubit is a vector representing the location of an electron between the ground and excited state, where ground state is the qubit  $|0\rangle$  and the excited state is the qubit  $|1\rangle$ .

A qubit is used to describe the peculiarities of quantum mechanics, which can be any of the following like,

- the spin of an electron where in the two spin levels are taken into consideration for qubit i.e. spin up and spin down
- the polarization of a photon where in the two states can be taken for qubit consideration as vertical polarization and the horizontal polarization

The notations  $|0\rangle$  and  $|1\rangle$  are designed to suggest a very helpful analogy with a normal digital bit (binary digit) taking the values 0 or 1.

A single qubit state is always a linear combination of base vectors  $|0\rangle$  and  $|1\rangle$ , given as :

$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$  wherein  $\alpha$  and  $\beta$  are complex numbers.

A state of a quantum computer having  $n$  qubits is a point in a Hilbert space with  $2^n$  orthogonal basis vectors, each of which is again a point in a complex 2-dimensional Hilbert space.

While a bit can either have a value 0 or 1, a qubit is actually a linear superposition of both a  $0$  and  $1$  value. Furthermore, a qubit can also be intertwined with other qubits, which is the primary reason for a quantum computer's remarkable computational strength.

A quantum gate is a qubit operation to change its state. We must have at least a two-qubit gate for example like a CNOT, to create entanglement.

Where a classical bit can take any one of the two states (zero or one), a quantum bit can take a not just logic 1 or logic zero but also a combination of both at the same time.

Fig 1 describes a comparison between a classical bit and a quantum bit.

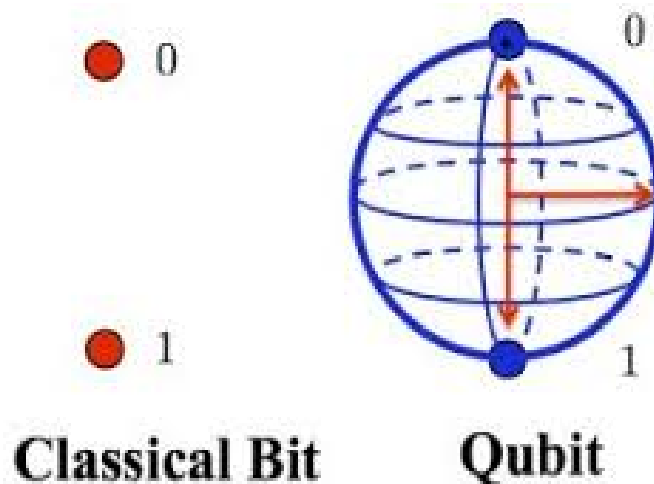


Fig. 1 Classical vs Quantum bit



## THE BLOCH SPHERE:

A Bloch sphere[14] is a standard Visualization tool for understanding what is happening in a quantum circuit. As a three-dimensional object, it reflects a qubit. Any state can be written as according to the equivalent representations of the states through the Bloch diagram.

$$|\psi\rangle = \cos\theta/2 \cdot |0\rangle + e^{j\phi}\sin\theta/2 \cdot |1\rangle$$

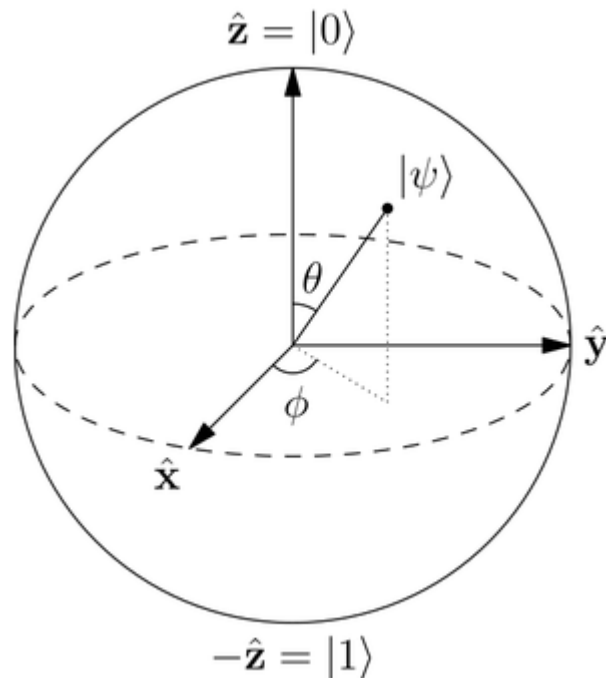


Fig.2 Qubit on a bloch sphere.

Fig. 2 defines a qubit on a bloch sphere wherein the dot represents a qubit, z axis defines the logic zero and logic one state of the qubit,  $\theta$  represents the latitude and  $\phi$  represents the longitude.

The range of values for  $\theta$  and  $\phi$  so that they cover the entire realm (without “repetitions”) is  $\theta \in [0, \pi)$  and  $\phi \in [0, 2\pi)$ . The angle  $\theta$  refers to the latitude and the angle  $\phi$  is the longitude. For example:

- Assume that  $\theta=0$ . This means that:  $|\psi\rangle = 1 \cdot |0\rangle + e^{j\phi} \cdot 0 \cdot |1\rangle = |0\rangle$ . Now assume that  $\theta=\pi$ ; similarly we get:  $|\psi\rangle = 0 \cdot |0\rangle + e^{j\phi} \cdot 1 \cdot |1\rangle = e^{j\phi} \cdot |1\rangle = |1\rangle$
- Assume that  $\theta=\pi/2$  and  $\phi=0$ . Then,  $|\psi\rangle = 1/\sqrt{2} \cdot |0\rangle + e^{j0} \cdot 1/\sqrt{2} \cdot |1\rangle = (|0\rangle + |1\rangle)/\sqrt{2}$ , while for  $\phi=\pi$ , we get  $|\psi\rangle = 1/\sqrt{2} \cdot |0\rangle + e^{j\pi} \cdot 1/\sqrt{2} \cdot |1\rangle = (|0\rangle - |1\rangle)/\sqrt{2}$

The Bloch sphere interprets as follows: :

The poles are classic bits, let's use the notation  $|0\rangle$  and  $|1\rangle$ . While these are the only possible states to represent the classical bit, quantum bits cover the entire sphere. In the quantum bits, therefore, there is much more data engaged, and that is depicted by the Bloch sphere.

## STATE VECTORS:

A quantum computer's state space with  $n$  qubits can be described as a tensor product " $\otimes$ " of all the individual qubits's respective state spaces.

Tensor Products are used to define multi-subsystem systems. Each subsystem in a vector space (Hilbert space) is defined by a vector. Let's have two schemes 1 and 2 with their Hilbert spaces  $H_1$  and  $H_2$  for instance. Hence on using the bra-ket notation (Bra-ket notation is one standard notation to represent quantum states, which uses a vertical bar and an angle bracket with the qubit in between to signify a quantum state) [11], the vectors  $|a_1\rangle$  and  $|a_2\rangle$  portray the system state 1 and 2 with the complete system state given by the tensor product  $|a_1\rangle \otimes |a_2\rangle$ .

For ease of notation, we abbreviate:

$$|a_1\rangle \otimes |a_2\rangle = |a_1a_2\rangle, \text{ etc.}$$

The two qubits tensor product over the bases  $|0\rangle$  and  $|1\rangle$  has the four bases  $|00\rangle$   $|01\rangle$   $|10\rangle$   $|11\rangle$ .

The tensor product follows the distributive law with vector addition, which can be explained with the example below:

Let us define two qubits  $X$  and  $Y$  defined as  $X = (a_1|0\rangle + b_1|1\rangle)$  and  $Y = (a_2|0\rangle + b_2|1\rangle)$ , hence for the two qubits  $X$  and  $Y$ , tensor product is given as:

$$(a_1 |0\rangle + b_1 |1\rangle) \otimes (a_2 |0\rangle + b_2 |1\rangle) = \\ a_1a_2 |00\rangle + a_1b_2 |01\rangle + b_1a_2 |10\rangle + b_1b_2 |11\rangle$$

### 3.4 PROPERTIES OF QUBITS:

#### QUANTUM ENTANGLEMENT:

Quantum entanglement[7] is one physical phenomenon which occurs whenever a pair or a group of particles such as electrons or photons are generated, in such a way that they are either interacting or sharing spatial proximity in ways like the quality of each particle can not be described independent of the other's state, but as a system whole, no matter if the particles are even separated by a big distance.

Measurement of physical characteristics such as spin, momentum, position, and polarization are said to be correlated which can be further explained by an example like for instance, consider a pair of particles which are generated in a way such that if one particle is vertically polarised then the other one is horizontally polarized(as shown in Fig. 3) then the two particles are said to be entangled. Another example of quantum entanglement could be like if a pair of particles is generated in a way such that their total spin is considered to be zero, and if one particle is known to have a clockwise spin on a certain axis, then owing to their entanglement property, the spin of the other particle is discovered to be counterclockwise, when measured on the same axis. However, this conduct provides rise to some evident paradoxical effects such that any measurement on a particle's property individually, which is entangled with another particle already, performs a collapse which is irreversible on that particle and hence altering its initial quantum state. Such a test will be done on an entangled system as a whole in the event of entangled particles.

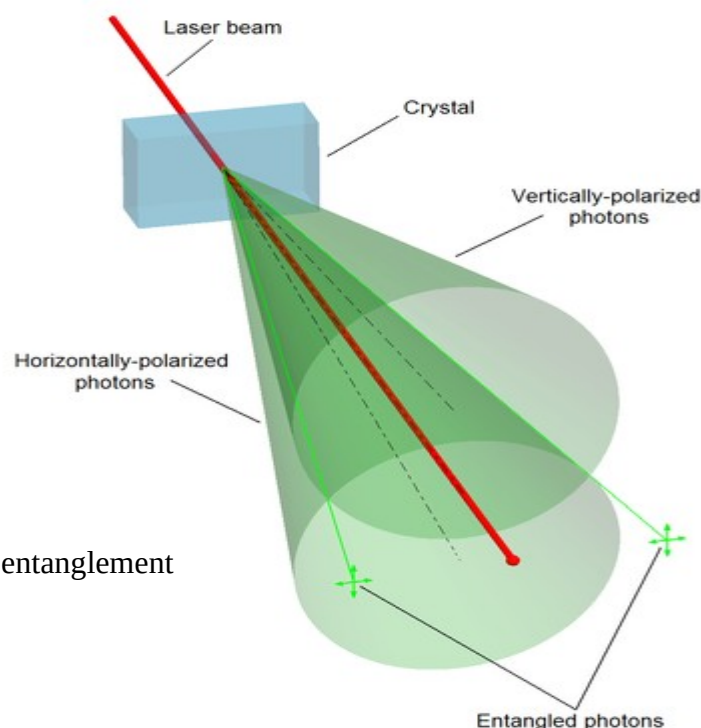


Fig. 3 Quantum entanglement

Entanglement is the strangest of

the quantum phenomena. Two or more quantum objects are entangled when they act in ways that are too distant to influence one another such that they are

1) individually random, yet

2) too closely linked

to be explained by supposing that each object is independent from the other.

A complicated or entangled state is a condition in which there are many qubits that can not be expressed as a list of different qubits. For example, none of these two-qubit states:  $|00\rangle$ ,  $|01\rangle$ ,  $|10\rangle$ , or  $|11\rangle$ , is entangled because they can all be described by attributing a definite state to each qubit.

The state  $(|00\rangle+|01\rangle)/\sqrt{2}$  can be expressed by saying the first qubit is in a superposed single-qubit state  $(|0\rangle+|1\rangle)/\sqrt{2}$  and the second in the  $|0\rangle$  state, so is not entangled. However, the state  $(|01\rangle+|10\rangle)/\sqrt{2}$  is entangled, because there is no way to express it as a list of single-qubit states. When measuring one of the qubits in this state, it acts randomly along any measuring axis at all, but its random behavior allows you to predict perfectly how the other qubit would behave when measured along the same axis. This unique mixture of ideal correlation with ideal individual randomness is displayed by no unentangled state.

## QUANTUM SUPERPOSITION[15]

A superposition is actually a weighted sum/difference of two or more states, such as the air condition when two or more musical tones sound at the same time. Superposition can be classical or ordinary, but in macroscopic phenomena both happen frequently, which involves waves.

How a quantum superposition is special? Here is how. Quantum theory predicts that in a superposition of all  $2^n$  of its separate logical states, a computer with  $n$  qubits can occur as a superposition of any combination of states from  $|000\dots 0\rangle$ , through  $|111\dots 1\rangle$ . This is more than a classic superposition exponentially. It's like playing  $n$  musical tones at once and they can all generate only one overlay of all  $n$  states.

If we compare this quantum superposition from classical probability, it is different in the way as described with the example as follows.

$N$  coins, each of which may be a head or a tail, can be described as a probabilistic mixture of  $2^N$  states, but in fact it's just one of those states — we just don't understand what. Quantum superposition is therefore much more potent than conventional probabilism. Quantum computers capable of superposing their information can solve issues at a pace that is exponentially quicker than any known classical deterministic or probabilistic computer. A more technical distinction is that while probabilities have to be positive (or zero), the superposition weights can be positive, negative, or even complex numbers. Fig. 4 illustrates a difference between a qubit and a classical

bit. Where a classical bit can take any one of the two states ( zero or one), a quantum bit can take not just logic 1 or logic zero but also a combination of both at the same time.

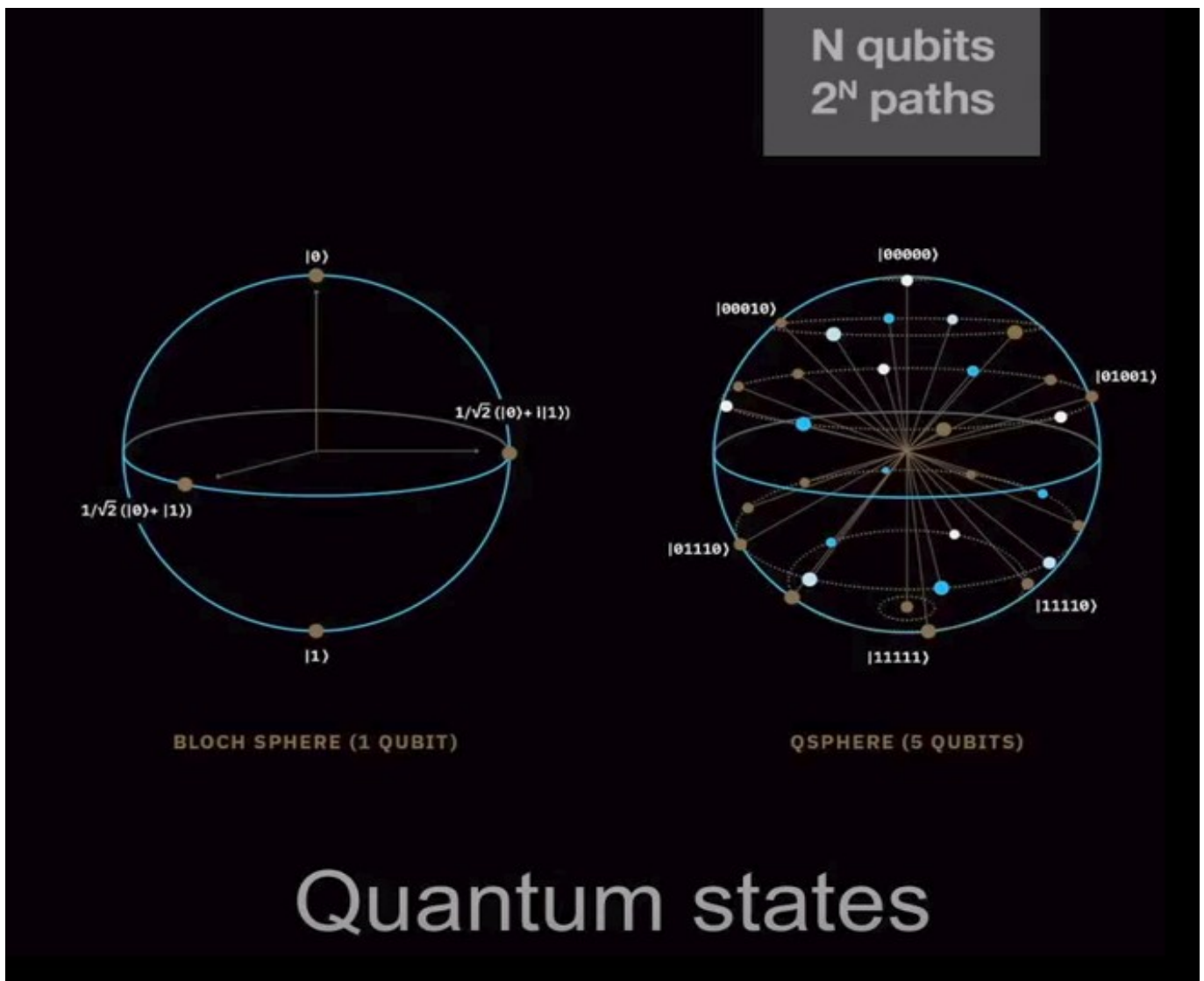


Fig. 4 Illustration of classical vs quantum states

### 3.5 PHYSICAL INTERPRETATION OF QUBITS :

The qubits used physically at IBM currently are superconducting transmon qubits. It's a qubit-based Josephson junction that's insensitive to charge noise. Compared to tunable qubits, these are fixed-frequency qubits to minimize their sensitivity to internal changes in the magnetic field that could corrupt quantum information.

The IBM manufactures these superconducting qubits. The instruments are produced of silicon wafers with superconductive materials such as niobium and aluminum that are preserved at superconductive temperature ( $\sim 10$  mili Kelvin) to generate a Josephson junction that is accountable for keeping the qubit in place. In a printed circuit board package, the quantum processor itself is housed . This unit is installed inside a light-tightened magnetic field shielding can that is located at the coldest point at the bottom of a dilution refrigerator that enables the qubit to function at a seperation rate of  $\sim 5$ GHz of energy, housed in one of IBM's Quantum Computing laboratories.

### 3.6 JOSEPHSON JUNCTION AND QUBITS:

A Josephson Junction is based completely on the superconducting quantum mechanical phenomenon. A ring of superconductive material broken into two locations by cables is the easiest model of a Josephson Junction(Fig 5.) A superconductor's most significant feature is that its a material having zero resistance and hence the superconductor's name. The result of this is that there is no energy dissipation because of collisions when a current is started in a superconductor.

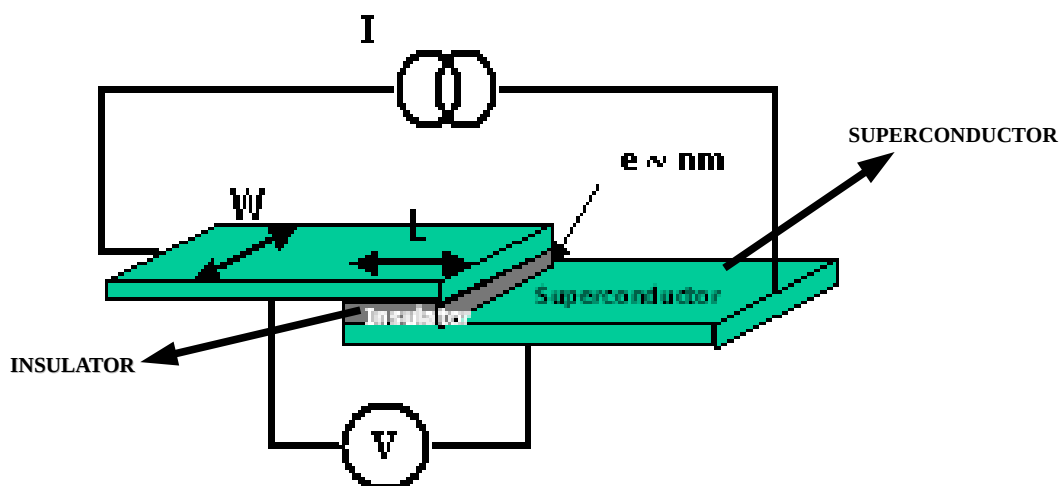


Fig. 5 Josephson junction model

Either Niobium or Aluminum is the type of materials used for superconduction. The electrons in a metal, pairs up to form Cooper pairs in some special superconductors, such as Aluminium and Niobium, for a particular critical temperature. Then, like ordinary electrons, these cooperating pairs form a current. The pairs follow each other in the loop, in the same way as a professional biker drafts to decrease drag impacts owing to wind, and hence the electrons end up travelling through the metal without facing any resistance, owing to energy gap between bonded and free electrons. To make superconduction a reality, our laboratory temperature must first be lowered. Typically, at 9.3 K and 1.2 K, Niobium and Aluminum become superconductors. There are few devices at these temperatures that can function as a circuit element. The superconducting Josephson Junction is actually the only circuit element that can function non-dissipatively. It turns out that if we make use of the Cooper pairs mentioned in the next section, we can use Josephson Junctions as a two tier structure. In order to use Josephson Junctions for qubits generation, we must somehow be able to introduce a two-level system generating a superposition(overlap) of eigen states ( $|0\rangle$  and  $|1\rangle$ ). The most evident way to do so is to use electron's eigen energies in superconducting state . Due to the energy dissipated by heat oscillations ( $kT$ ), we have to restrict our temperature to such a range that  $kT < \omega_0 \hbar$ . The frequency of oscillations of superconducting qubits typically ranges from 5 - 20 GHz, which makes it necessary to enforce the qubits at a temperature of around 20 mK. Using present laboratory techniques, this low temperature can be achieved. Once we achieve this low temperature, in conjunction with capacitors and resistors, we can use Josephson Junctions to create qubits to create quantum computers.

## **COOPER PAIRS , JOSEPHSON JUNCTION AND QUANTUM CURRENTS**

A variable flux through a conductive loop produces currents in the loop according to classical electromagnetics. This phenomenon is reflected in a Josephson Junction superconduction situation. When we deal with the quantum system, however, we cannot actually take the currents as true amounts, but rather as wave functions with probability distribution. With the help of such a view regarding current and charge, we can have a situation where a loop current moves simultaneously for both the input and output directions, since a wave function can be a superposition of many current states, each having different likelihood of getting measured for a given time instance.

The development of superconducting pairs is due to the phenomenon of electrons exchanging phonons with each other at low temperatures in superconductors. This exchange creates a bond between the two electrons in a Cooper pair. They become a boson instead of two fermions i.e. an electron with a spin of 0, when these electrons bind together. From the concept of the quantum field theory, we understand that Bosons can be all placed in a metal's ground state, as opposite to fermions that must follow the principle of exclusion of Pauli. This is the origin of superconduction due to the fact that electrons in a Cooper pair possess an

energy gap in the bound state which do not have enough energy to come out of the metal lattice and therefore experience zero resistance when traveling through the metal. So we have quantum currents that flows without dissipating any energy, a main criterion for maintaining quantum qubit consistency.

## SUPERCONDUCTING QUBIT DEVICES:

There are several, to be specific, three ways to utilize superconduction for creating a qubit:

1) The first way exploits the “charge qubits” of the cooper pair and could be viewed as being equal to a quantum circuit hydrogen atom. We can easily use the lowest possible non-degenerate state as a "charge qubit." Fig. 6 shows a cooper pair box diagram representing a josephson junction with cooper pair formation .

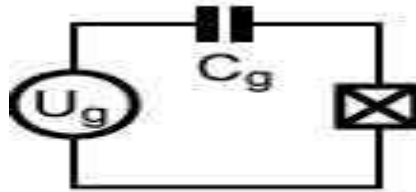


Fig 6 .A simple cooper pair box diagram

2) The Radio Frequency Superconducting Quantum Interference Device is the second device that utilizes superconduction to introduce qubits. As the name suggests, it is often used in Radio Astronomy applications and also its characteristics are prominently recognized so. It uses a couple of inductors to detect the flow through the loop (Fig 7). This flux in turn produces a current that can then be used to produce a qubit. The presence of a degenerate ground state characterizes this circuit. The energy between the two degenerate ground states can be splitted and adjusted by merely altering the bias current across the intersection and using this split level as a qubit.

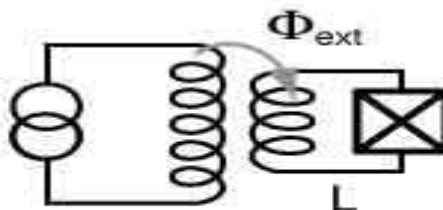


Fig 7. Diagram of a SQUID device. The inductor induces a current in the second loop.



3) The third and final developmental superconducting device is the current Josephson Junction (fig 8). The splitting of energy within the potential wells is non-degenerate. This fact makes this instrument highly beneficial. This fact along with the fact that bias vs. potential current is a "tilted washboard" shape, this non-degenerate splitting means that the first excited state is  $|1\rangle$ , if we treat the ground state of the well as a  $|0\rangle$  and, so we can easily determine the state of the electron by sending a microwave pulse if we are uncertain whether or not it is in  $|0\rangle$  or  $|1\rangle$ . In case we are in a  $|1\rangle$  state, it is 500 times more probable to cause penetration of barriers than the case if we were in  $|0\rangle$ . In addition, due to the prospective form of the washboard, an electron that passes through the obstacle will then move down the potential ultimately manifesting itself through the Josephson Junction as a current spike. Hence, in case our pulse outputs a voltage, we now know an electron has been excited from the  $|1\rangle$  state and not  $|0\rangle$  state.

This third type is the type of superconducting qubits being in practice at the IBM for the quantum computer.

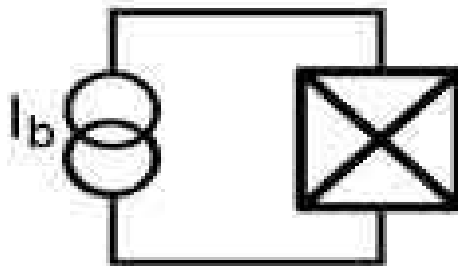


Fig 8. Current biased josephson junction

## HOW QUANTUM GATES ARE PERFORMED PHYSICALLY:

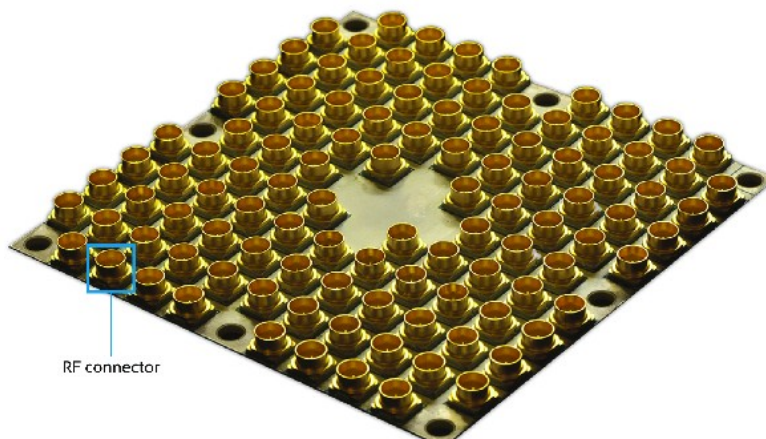
Quantum gates are physically conducted by sending electromagnetic impulses through coaxial cables at microwave frequencies to the qubits. These electromagnetic pulses have a specific phase, frequency, and duration that determines the rotation angle of the qubit state around a particular Bloch sphere axis.

A Bloch sphere is a standard tool for the visualization of what happens in a quantum circuit. As a three-dimensional object, it reflects a qubit. Typically, two-qubit gates require microwave frequency pulse tuning to calibrate the interaction between the two qubits during a gate duration and minimize the interaction at any other time. Since our choice qubits are fixed-frequency transmons, we can not adjust the interaction by getting them closer to a two-qubit gate in terms of frequency. Rather, the property utilized for this interaction is the cross-resonance impact, driving one of the qubits (called control) with a microwave pulse tuned at the second qubit frequency (called target). This phenomenon can actively enhance the coupling strength between them. The nature of the cross-resonance impact also enables us to rotate in the destination qubit conditioned on the control qubit status, a main feature of the CNOT operation needed for the universal quantum gate set.

# INTEL'S 49-QUBIT PROCESSOR

During his keynote at CES 2018 in January, Intel CEO Brian Krzanich unveiled our 49-qubit superconducting quantum test chip, code-named "Tangle Lake." The 3-inch by 3-inch chip and its package is now in the hands of Intel's quantum research partner QuTech in the Netherlands for testing at low temperatures. Quantum computing is heralded for its potential to tackle problems that today's conventional computers can't handle. Scientists and industries are looking to quantum computing to speed advancements in areas like chemistry or drug development, financial modeling, and even climate forecasting.

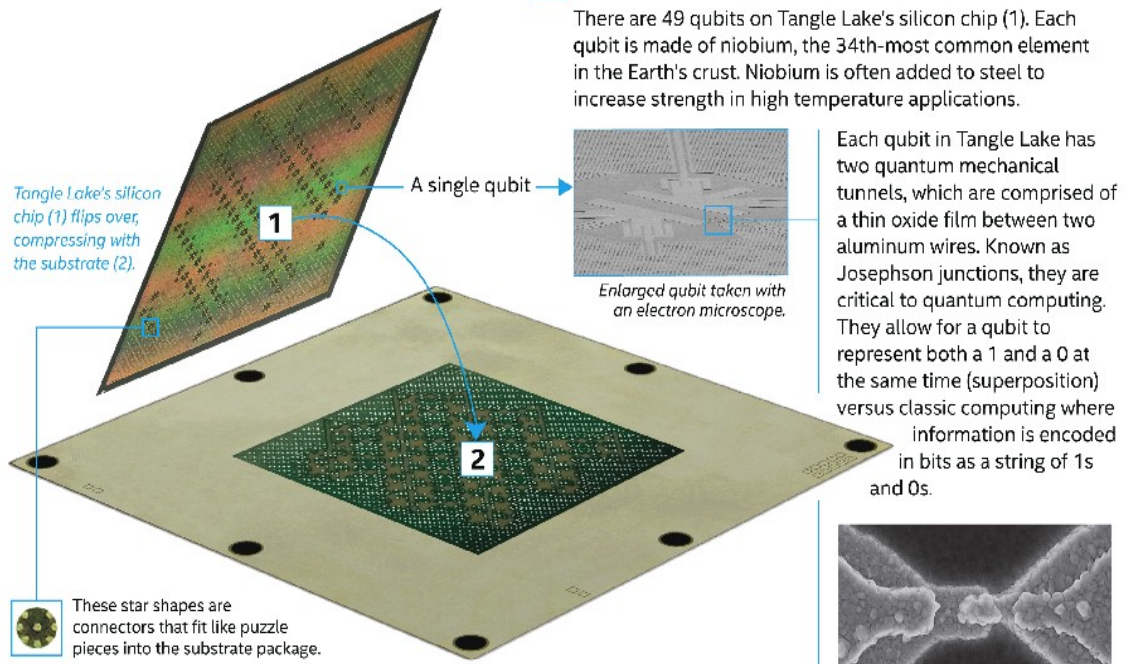
TOP



## WORTH ITS WEIGHT IN GOLD

There are 108 radio frequency (RF) connectors on Tangle Lake that carry microwave signals into the chip to operate the quantum bits (qubits). They are made of gold, which is excellent for anti-corrosion and signal transmission.

THE MAGIC INSIDE



CONNECTING THE QUBIT CHIP

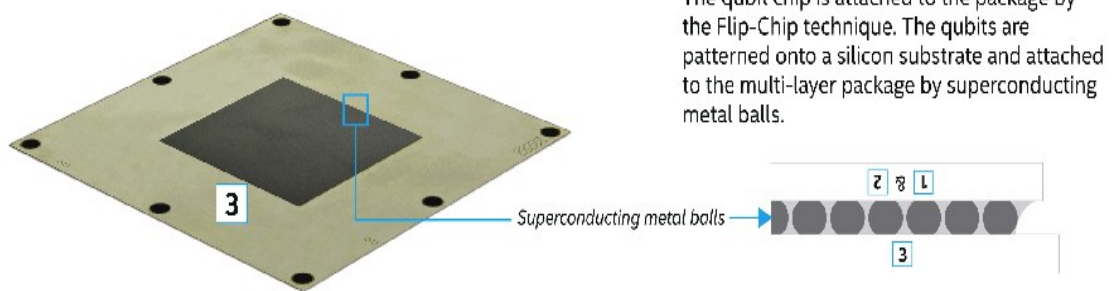


Fig. 9 The latest INTEL's 49 qubit quantum processor which uses current biased josephson junction for Qubits.

Fig 9 represents the latest INTEL 49 qubit processor "Tangle Lake"[16] which processes 49 qubits at a time. To process the Qubits , the third type of josephson junction defined in the previous section i.e. the current biased josephson junctions are used ,made up with Niobium.

### **3.7 DIFFERENCES BETWEEN CLASSICAL AND REVERSIBLE COMPUTING:**

Understanding the differences from the synthesis point of view between non-reversible circuits and reversible circuits is of utmost importance to get through with reversible synthesis of logic. The distinctions between reversible logic synthesis and binary logic synthesis differ substantially with the synthesis's mathematical and algorithmic elements. The differences can be summarized as follows:

1) While conventional digital logic gates have many inputs and one or two outputs (or an unequal amount of inputs and outputs), the gates used in most reversible logic systems (particularly in quantum computing) have equal input and output signals. In this dissertation, we use this gate model, called  $k \times k$  gate.

2) Not every output of a reversible gate is helpful and it is called a trash output (bit, qubit, etc.) that is not used to provide data to other gates in the circuit or is not measured in quantum for further procedures. In non-quantum techniques, garbage waste energy. In quantum technology, they waste computing resources, so is their name. They also waste power when measured in quantum computing. The amount of trash inputs should be minimized by an excellent synthesis technique. Minimizing the amount of waste production (garbage qubits) in quantum computing is one of the primary synthesis criteria.

3) Arbitrary logic function can be transformed to a reversible function by adding to Boolean constants a tiny amount of extra bits initialized (0 or 1). These are called ancilla bits, and the synthesis objective is to keep as low as possible the number of ancilla bits.

4) Any gate output in a reversible circuit can only be used once (the fan-out for a reversible gate has to be mandatorily one). If a reversible circuit requires copies of a particular signal, a circuit capable of should be used for this case. This is performed in quantum because the technology does not allow circuit fanouts. This copying circuit is a "Toffoli gate" with the control bit set to the copying signal and set to zero, the target bit. The circuit qubit width must be improved by one for each single copy of the output signal in order to integrate fanout of more than one for a specific signal.

5) In reversible circuits, the reversible circuits arising from their synthesis are fundamentally acyclic, which is not permitted in feedback loops. The design of sequential circuits or combination circuits with loops, with reversible gates, is therefore a broad area of studies as sequential circuits require feedback loops to generate memory elements.

## CHAPTER 4 - Simulation and Synthesis of reversible logic circuits

### 4.1 FAMILY OF BASIC REVERSIBLE GATES :

#### 1) CNOT GATE:

The controlled NOT gate (also known as CNOT) is an quintessential component in the building of a gate-based quantum computer. It can be used to intertwine and disassemble any of the physical properties of particles such as the Qubit's position, momentum, spin and polarization. A mixture of single qubit rotations and CNOT gates can be used to simulate any quantum circuit to an arbitrary degree of precision. The CNOT gate works on a 2-qubit quantum register, and only if the first qubit (also known as the control qubit) is one, the CNOT gate flips the second qubit (also known as the target qubit).

If only one and zero are allowed for the target and control bit, the target bit output represents an XOR of the input control and target bits.

Tab 1 – CNOT Gate

INPUTS		OUTPUTS	
A	B	P	Q
0	0	0	0
0	1	0	1
1	0	1	1
1	1	1	0

#### 2) TOFFOLI GATE:

In logic circuits, a real reversible logic gate is the Toffoli gate (also called the CCNOT gate), created by Tommaso Toffoli. This implies that from Toffoli gates any reversible circuit can be built. It is also known as the "controlled-controlled-not" gate, which also vividly describes its action. It has 3 input bits and output bits ; if both the first and the second bits are set to one, the third bit will be inverted, otherwise all bits remains the same.

For a single bit input, there are two possible reversible gates. One of them is the NOT gate. The other is the gate of identity, which maps its input in an unchanged fashion to the output. The only non-trivial gate for two input bits is the controlled NOT gate, which XORs the two bits i.e. the target and the control bit.

Table 2 – Toffoli Gate

INPUTS			OUTPUTS		
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

Any reversible gate can be implemented on a quantum computing, and hence the Toffoli gate is also a quantum operator.

### 3) FREDKIN GATE:

The Fredkin gate (also known as the CSWAP gate) is a reversible computing system invented by Edward Fredkin. . It is also universal, means that any arithmetic or logical operation can be built using only Fredkin gates . The Fredkin gate is a circuit or device with 3 input bits and 3 output bits that transmits an unchanged first bit and swap the second and third bits only if the first bit is 1.

Table 3 – Fredkin Gate

INPUTS			OUTPUTS		
C	I1	I2	C	O1	O2
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

Universality of the Fredkin gate can be discovered from the fact that it can be used to implement all the for logical and arithmetic operations i.e. AND, XOR, OR and ADDITION.

#### 4) FENYMAN GATE:

Fenyman gate is a 2\*2 reversible gate where one of the output is same as the input(garbage) and the other output is an XOR of the two inputs.

Table 4 – Fenyman Gate

INPUTS		OUTPUTS	
A	B	P	Q
0	0	0	0
0	1	0	1
1	0	1	1
1	1	1	0

It is possible to use Feynman Gate (FG) as a copy gate. Since fan-outs are not permitted in reversible logic, this gate is helpful to duplicate the required outputs by setting one bit to one and thus the fenyman gate's XOR output will provide a copy of the other bit.

#### 5) PERES GATE:

The peres gate is a 3\*3 reversible gate. The output is defined by  $P = A$ ,  $Q = A \oplus B$  and  $R = AB \oplus C$ . A Peres gate's quantity cost is 4. Most of the reversible circuits make use of Peres gates to the maximum. The reason behind this is because peres gate has the lowest quantum cost when compared to all other 3 input reversible gates.

Table 5 – Peres Gate

INPUTS			OUTPUTS		
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	0
1	0	1	1	1	1
1	1	0	1	0	1
1	1	1	1	0	0

## 4.2 SYNTHESIS OF REVERSIBLE GATES:

The synthesis of reversible circuits is a wide area of research and is of much importance because it gives the insight into the practical implementation of a reversible circuit.

In general, the synthesis methods target an optimised and efficient realization from the reversible specification. Various algorithms are applied to convert any reversible specification into a network of Toffoli gates. The prime focus of these methods is to minimize the gate count and the quantum cost in the implementation. The synthesis technique used in this dissertation work is the truth table based synthesis technique for reversible circuits.

The technique is described as follows:

### REVERSIBLE SYNTHESIS BASED ON TRUTH TABLE TRANSFORMATIONS :

Synthesis techniques based on truth table are based on truth table transformations. A sequence of transformation iterations is applied during a reversible logic synthesis method and a sequence of Toffoli gates implements the entire method.

There are three ways of applying Toffoli gates to a reversible function specification:

- a) Reversible function inputs result in synthesis flow from input to output
- b) Synthesis flows from output to input because of Toffoli gates application to output of a reversible function
- c) Toffoli gates applied to both inputs and outputs of a reversible function resulting in a simultaneous synthesis flow in both directions.



In this approach , we use three types of toffoli gates for the Toffoli network;

- TOF1 (X) known as NOT gate shown in Fig. 10,
- TOF2 (X:Y) known as CNOT gate shown in Fig. 11
- TOF3 (X, Y : Z) known as Toffoli gate shown in Fig. 12.



Fig. 10



Fig. 11

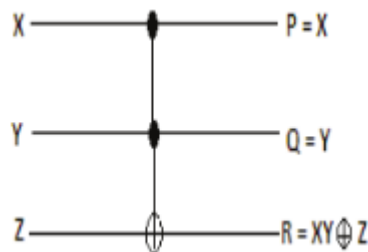


Fig. 12

## BIDIRECTIONAL SYNTHESIS ALGORITHM:

At first, a truth table is generated for a reversible function.

Let us describe the Bidirectional algorithm[8] for synthesis with an example.

- For our example reference let us say we have c,b,a as inputs and  $c^0 b^0 a^0$  as output without any transformation.
- We use the Bidirectional synthesis method for synthesis that synthesizes the circuit in both directions at the same time as gates are recognized to be added to either the input or output side of the network based on the benefit of saving gates after each transformation iteration.
- After each transition, this algorithm requires the intelligent choice to alter input or output based on whichever provides the greatest benefit in terms of gate savings.

For our example, the truth table representation is shown in Table

Table 6 Truth table for synthesis reference

INPUTS			OUTPUTS		
c	b	a	c <sup>0</sup>	b <sup>0</sup>	a <sup>0</sup>
0	0	0	1	0	1
0	0	1	0	1	0
0	1	0	0	0	0
0	1	1	1	0	0
1	0	0	1	1	1
1	0	1	0	1	1
1	1	0	0	0	1
1	1	1	1	1	0

According to the algorithm for synthesis, its visible that  $f(0) = 5$  and  $f(2) = 0$  so to transform the output bit from 101 to 000 leads to a change of 2 bits and to transform 010 to 000 leads to a change of 1 bit, which saves on the number of gates used for synthesis.

So the transformation is considered to be applied on the input side using Toffoli gate TOF1(b) and hence the output side transforms to the maxterms {2,3,0,1,6,7,4,5}.

Now on sorting the inputs in an ascending order, the output sequence gets reflects as {0,4,5,2,1,6,7,3}, which is shown as  $c^1 b^1 a^1$ . The process keeps on repeating till we get the transformed output bits  $c^7 b^7 a^7$  as shown in the table 7 below.

Now no more transformation is needed since all the bits have got their rightful positions.

Table 7 – Bidirectional synthesis algorithm

<b>cba</b>	<b>c<sup>0</sup>b<sup>0</sup>a<sup>0</sup></b>	<b>c<sup>1</sup>b<sup>1</sup>a<sup>1</sup></b>	<b>c<sup>2</sup>b<sup>2</sup>a<sup>2</sup></b>	<b>c<sup>3</sup>b<sup>3</sup>a<sup>3</sup></b>	<b>c<sup>4</sup>b<sup>4</sup>a<sup>4</sup></b>	<b>c<sup>5</sup>b<sup>5</sup>a<sup>5</sup></b>	<b>c<sup>6</sup>b<sup>6</sup>a<sup>6</sup></b>	<b>c<sup>7</sup>b<sup>7</sup>a<sup>7</sup></b>
000	101	000	000	000	000	000	000	000
001	010	100	101	001	001	001	001	001
010	000	101	100	100	010	010	010	010
011	100	010	010	010	100	011	011	011
100	111	001	001	101	101	101	100	100
101	011	110	111	011	011	110	111	101
110	001	111	110	110	111	111	110	110
111	110	011	011	111	110	100	101	111
Toffoli gate	T(b)	T(c:a)	T(a:c)	T(b:a)	T(c,a:b) T(a,b:c)	T(c:a)	T(c,a:b)	T(c,a:b)
Transformation Direction	←	→	→	←	← ←	→	→	→
INPUT/ OUTPUT	INPUT	OUTPUT	OUTPUT	INPUT	INPUT	OUTPUT	OUTPUT	OUTPUT

Toffoli network build using the above mentioned bidirectional algorithm is as shown in the Fig. below. It uses 8 toffoli gates in which 4 gates are 1 control toffoli gates,3 gates are 2 control toffoli gates and 1 gate without any control which is also called the CNOT gate.

This quantum equivalent circuit(Fig. 13) using bidirectional algorithm for reversible synthesis has been simulated and implemented on the tool QUIRK[12].

QUIRK is an open source web based drag and drop quantum circuit simulator which reacts, simulates and animates quantum circuits in real time.

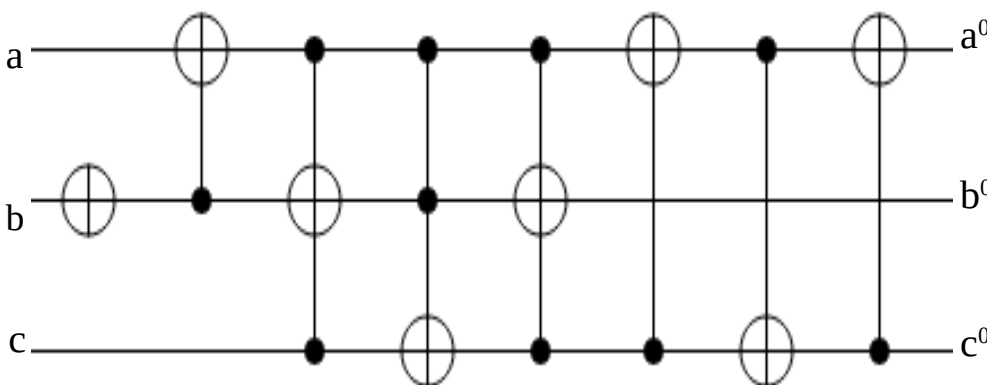


Fig. 13 Quantum equivalent

### 4.3 REVERSIBLE GATES WITH THEIR QUANTUM EQUIVALENT:

Following the truth table based synthesis technique, the quantum equivalent circuits of the basic reversible gates described in the previous section are as follows:

#### 1) CNOT GATE:

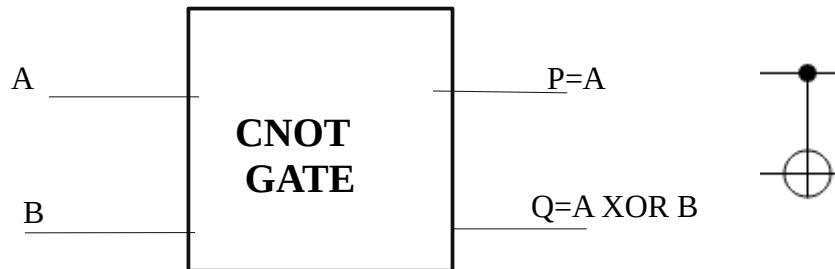


Fig 14 Reversible gate and its Quantum equivalent

#### 2) TOFFOLI GATE:

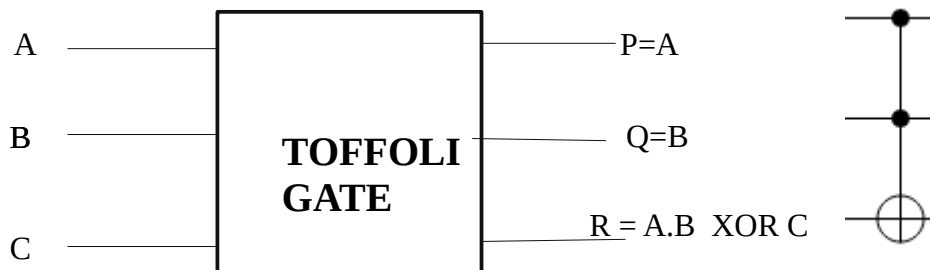
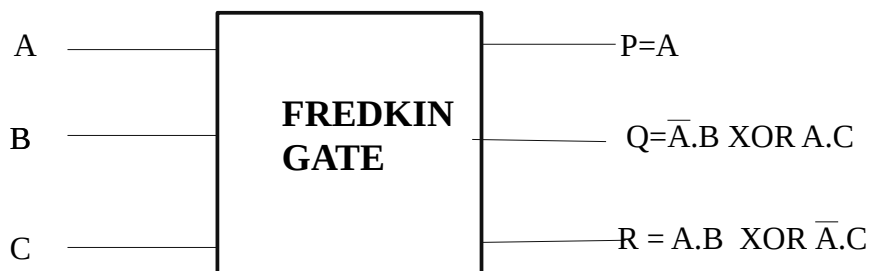


Fig. 15 Reversible gate and its Quantum equivalent

#### 3) FREDKIN GATE



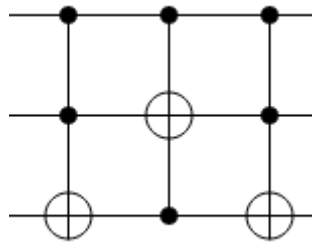


Fig. 16 Reversible gate and its Quantum equivalent

#### 4) FENYMAN GATE

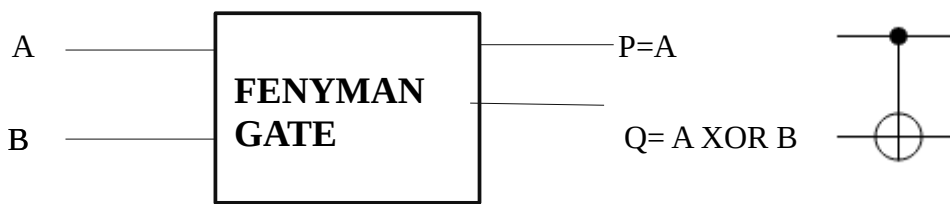


Fig. 17 Reversible gate and its Quantum equivalent

#### 5) PERES GATE

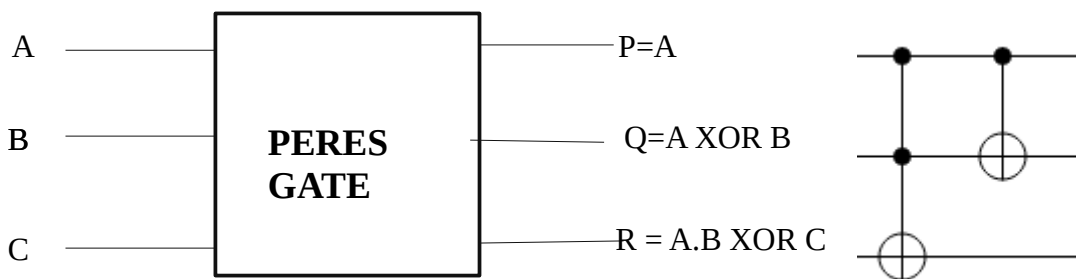


Fig. 18 Reversible gate and its Quantum equivalent

## CHAPTER 5 – Proposed Work and Results

### 5.1 REVERSIBLE ALU CIRCUIT DESIGN AND SIMULATION:

The following work on reversible circuits has been simulated on the open source quantum simulator QUIRK.

Before QUIRK, another open source C++ quantum library based reversible circuit simulator and synthesizer called Revkit[16] was attempted to be employed for reversible circuit synthesis. This tool is an open source toolkit for reversible circuit design. Revkit provides the basic functionality for reversible circuit design like Synthesis, Optimization, Verification. All the above functionalities and the involved algorithms are written in C++. Furthermore, to enable command line interface, all the functions are written and exposed as a python library.

However, currently a newer version of this tool is available for use but it is still in the development phase before being able to synthesize reversible circuits properly. An older version of the same which was in proper function mode some months back is also available to be used but is now no longer maintained by the tool owner hence making it impossible to make use of Revkit tool for reversible circuit synthesis.

The reversible ALU has been implemented by two Methods. In the first one, the ALU is synthesized by making use of different input combinations to a combination of PERES gate, FENYMAN gate and FREDKIN gate to get the desired functionality. However the number of ancilla bits and garbage outputs is too high for this implementation. Hence another Method is used in the next implementation for an N-Bit ALU which has a significantly lesser number of ancilla bits and garbage output bits. Using the Method II, a 2-bit, 4-bit and 6-bit ALU is created in this dissertation work.

The two Methods for synthesis of a reversible ALU is described as follows:

#### 5.1.1 METHOD I

This ALU is very easy to design and verify with this technique because it consists of several autonomous sub circuits. Being parallel, it is definitely quick, but at the cost of a big logic width and a big amount of continuous inputs, this speed is accelerated. In this Method[9], 3 basic reversible gates i.e. Peres, Fredkin and Feynman gates are used. Also called the Feynman gates is a copying gate that copies a little to be implemented as an input to the next door. This ALU uses 9 gates in total namely 4 Fredkin, 3 Peres, and 2 Feynman, which produces a quantum cost or quantum depth of 34.

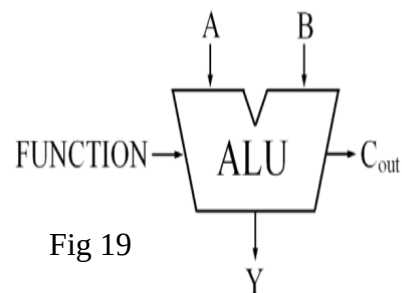


Fig 19

On performing a traditional logic calculation on this ALU realization yields that the hardware needed for this type of implementation comprises of 19 AND, 16 XOR and 8 NOT gates. The garbage/unused outputs are 6 and constant/ancilla inputs are 5.

A conventional(non reversible) ALU is a parallel combination of various arithmetic and logical operations using a multiplexer. Depending on the select line, the activities are carried out in multiplexer-based ALU .A straightforward ALU in its fundamental form comprises of two operand inputs, one operand selection input and one outcome output as shown in Fig. 19 and Fig. 20 . ALU's complexity may differ between processor and processor.

This is shown in the figure below:

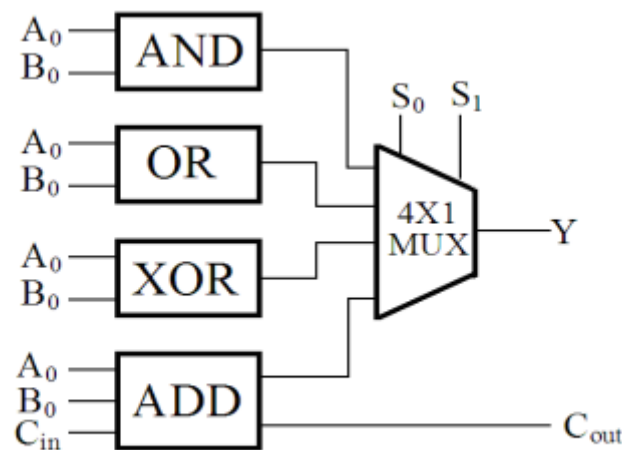


Fig 20 TWO BIT CONVENTIONAL ALU

2 bit ALU operation table :

The operation table for the above ementioned 2 bit ALU is given as below. The ALU performs the functions like OR, XOR, AND and ADDITION.

Table 8 – Operation table for a 2 bit ALU

INPUTS		OUTPUT
S0	S1	
0	0	AND(f0)
0	1	OR(f1)
1	0	XOR(f2)
1	1	ADD(f3)

The boolean equation for the outputs is as follows:

$$Y = \overline{S_0S_1}f_0 + \overline{S_0}S_1f_1 + S_0\overline{S_1}f_2 + S_0S_1f_3$$

$$C_{out} = S_0S_1(C_{in}(A \text{ XOR } B) + AB)$$

In the above expressions the values for functions f0,f1,f2,f3 is as below:

- f0= A.B
- f1= A+B
- f2= A XOR B
- f3= A XOR B

To achieve the above functionality, the building blocks of the ALU are created as given below:

**BUILDING BLOCKS OF A TYPICAL ALU USING REVERSIBLE GATES:**

1) AND GATE: An AND gate is implemented using the PERES gate as shown in the below figure.

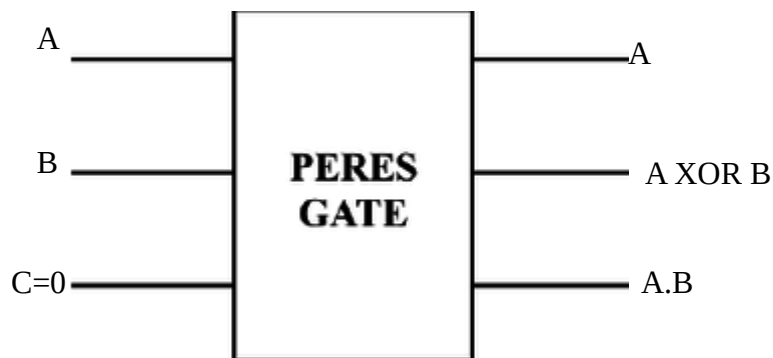


Fig 21 Reversible AND gate



2) XOR GATE: The XOR gate is implemented using the PERES gate by setting one input bit to a constant zero as shown in below figure.

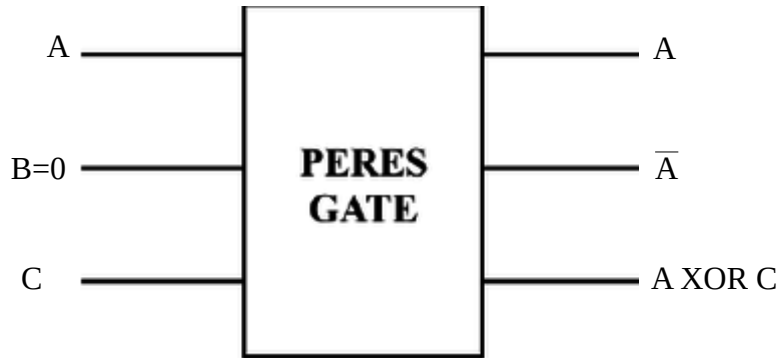


Fig 22 Reversible XOR gate

3) ADDER :

An adder is implemented using a combination of PERES gates , with one ancilla bit and two garbage outputs as shown below in the figure(Fig. 23):

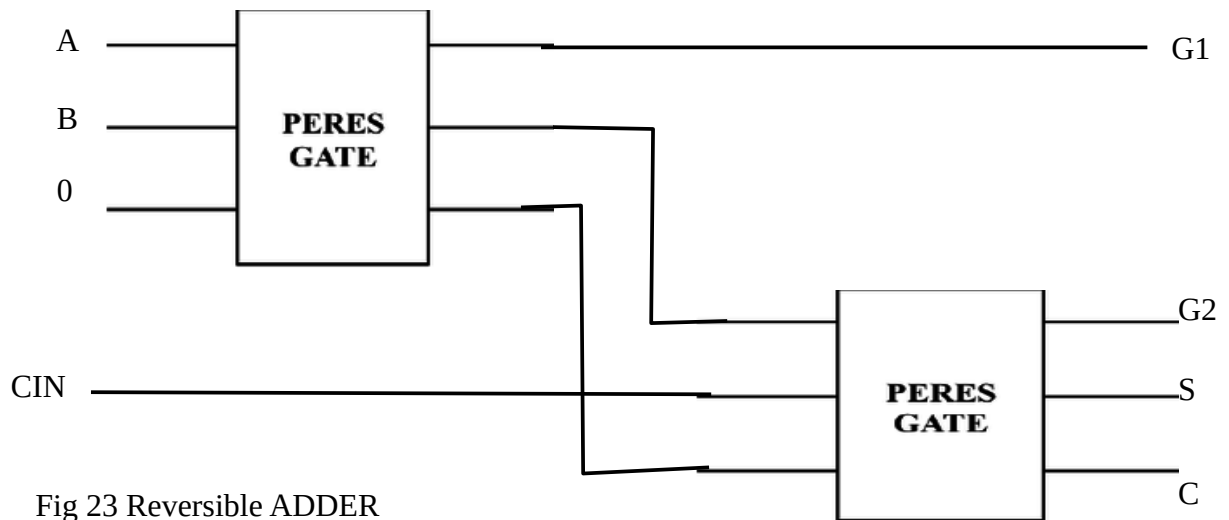


Fig 23 Reversible ADDER

### Operation Table of a 2 Bit ALU:

Table 9 - Operation table for a 2 Bit ALU

INPUTS		OUTPUT FUNTIONS
S0	S1	
0	0	AND(f0)

0	1	OR(f1)
1	0	XOR(f2)
1	1	ADD(f3)

The boolean equation for the outputs is as follows:

$$Y = \overline{S_0}S_1f_0 + \overline{S_0}S_1f_1 + S_0\overline{S_1}f_2 + S_0S_1f_3$$

$$C_{out} = S_0S_1(C_{in}(A \text{ XOR } B) + A.B)$$

In the above expressions the values for functions f0,f1,f2,f3 is as below:

$$f_0 = A.B$$

$$f_1 = A+B$$

$$f_2 = A \text{ XOR } B$$

$$f_3 = A \text{ XOR } B$$

Fig. 24 presents a 2 bit ALU constructed with a network of PERES gates , FREDKIN gaets and FENNYMAN gates.

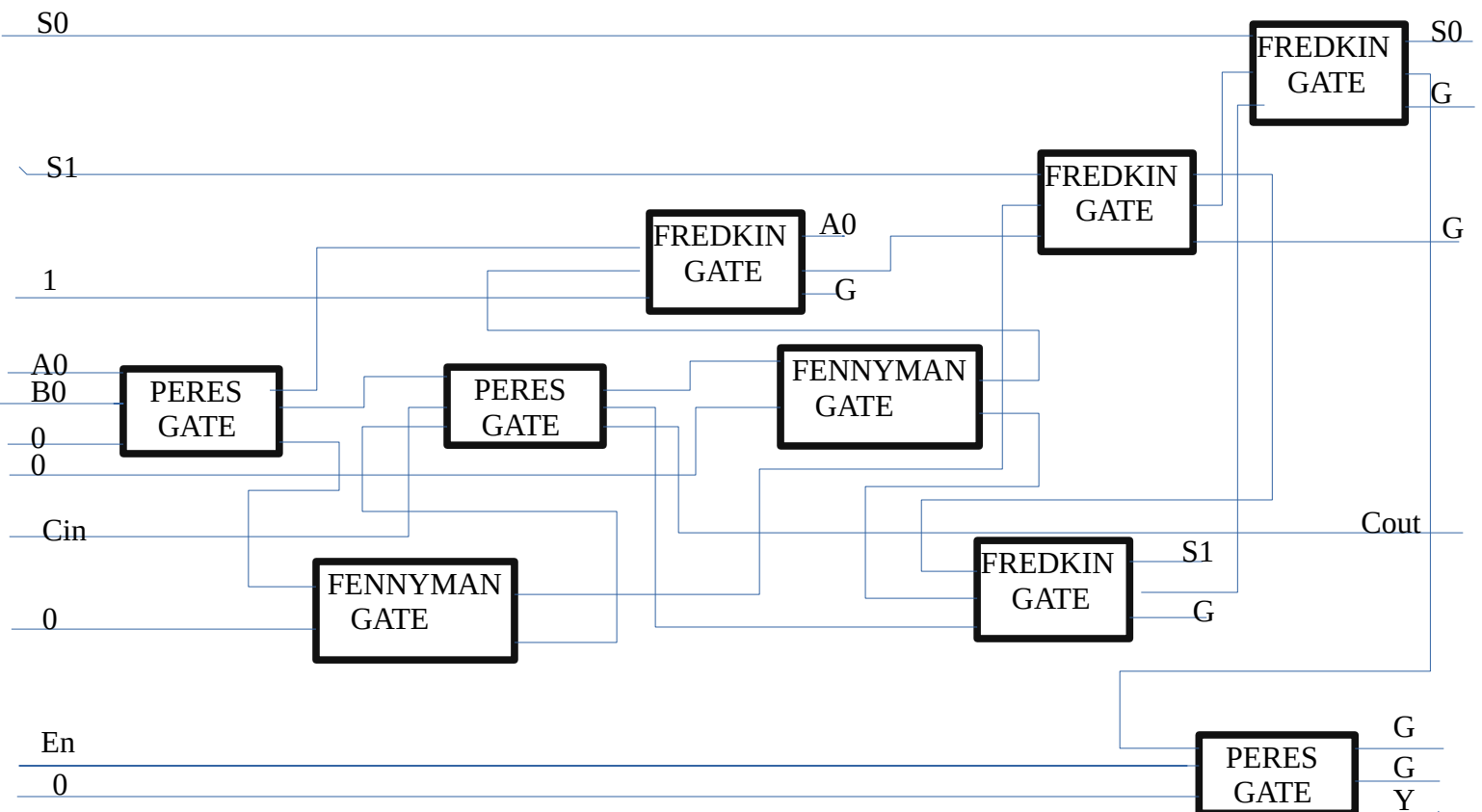


Fig.24 Reversible 2 bit ALU

The above reversible gates are converted to their respective quantum equivalent and the final circuit is then implemented on QUIRK for simulation and validation(Fig. 25).

The quantum implementation of the above ALU circuits is simulated on Quirk as shown below:

The outputs bits are signified with R0 and Cout.G signifies the garbage output bit. The input bits comprises of the select pins S0,S1 with 1 bit inputs A and B, carry input Ci and 5 ancilla bits .

Output bits are defined as:

$$R0 = A0 \diamond B0$$

$$Cout = Carry\_output ( A + B + Cin)$$

Here," $\diamond$ " represents a logical/arithmetic operation.

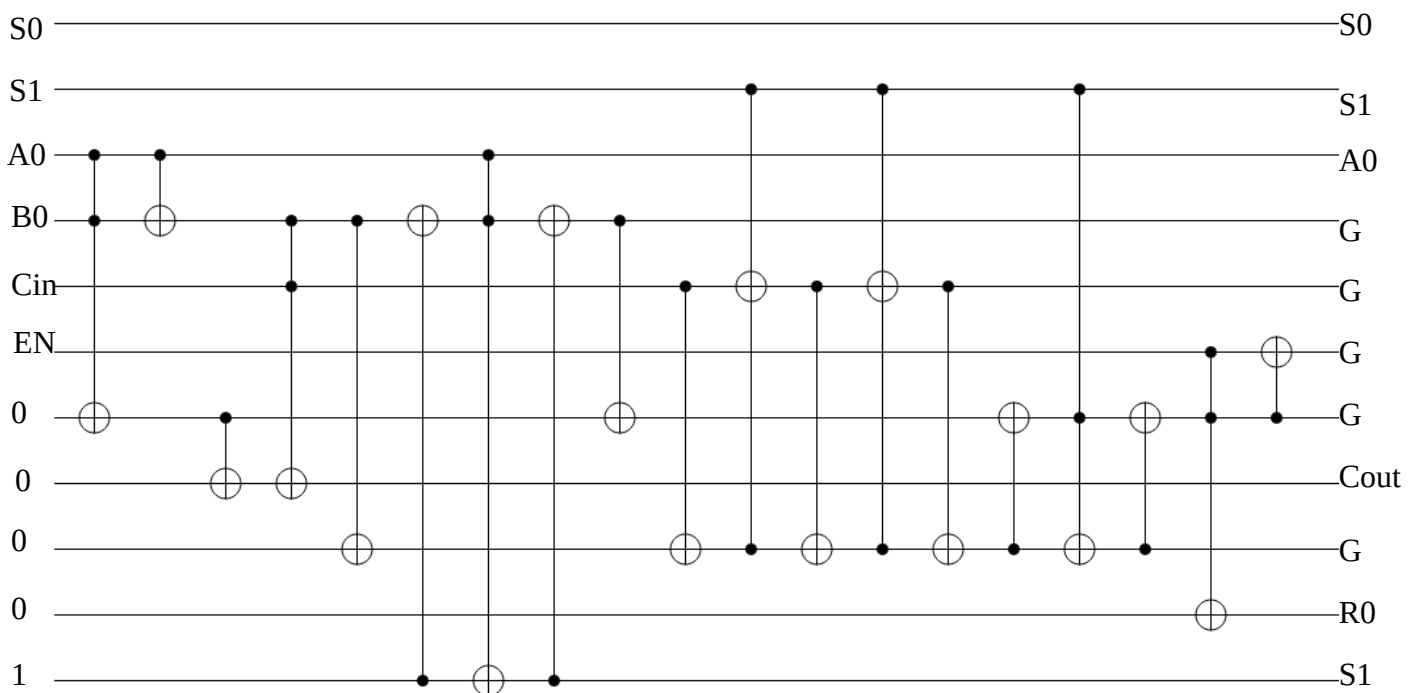


Fig 25 Quantum representation 2 bit ALU using Method I

### 5.1.2 METHOD II

The ALU implementation using the Method I has got a large number of ancilla inputs and garbage inputs. To be precise, a 2 bit ALU contains 5 ancilla input bits and 6 garbage output bits.

So this new Method, which is based on building individually, the quantum modules for the ALU and then multiplexing them all using the select inputs, creates a multibit ALU which is much better in terms of ancilla inputs and garbage outputs and hence the quantum cost as well.

Using the above mentioned Method to implement a 2 bit ALU with the similar functionality as defined in the operation table for Method I's 2 bit ALU.

The operation table is given as :

Table 10 - Operation table for a 2 bit ALU

INPUTS		OUTPUT FUNCTIONS
S0	S1	
0	0	AND
0	1	OR
1	0	XOR
1	1	ADD

The quantum implementation for the above functionality using Method II is given as follows(Fig. 26). The Output bit R0 and Cout is the arithmetic and logical operation output bit with 4 garbage output bits.

$$R0 = A0 \diamond B0$$

$$Cout = \text{Carry\_output} ( A + B + Cin)$$

Here," $\diamond$ " represents a logical/arithmetic operation.

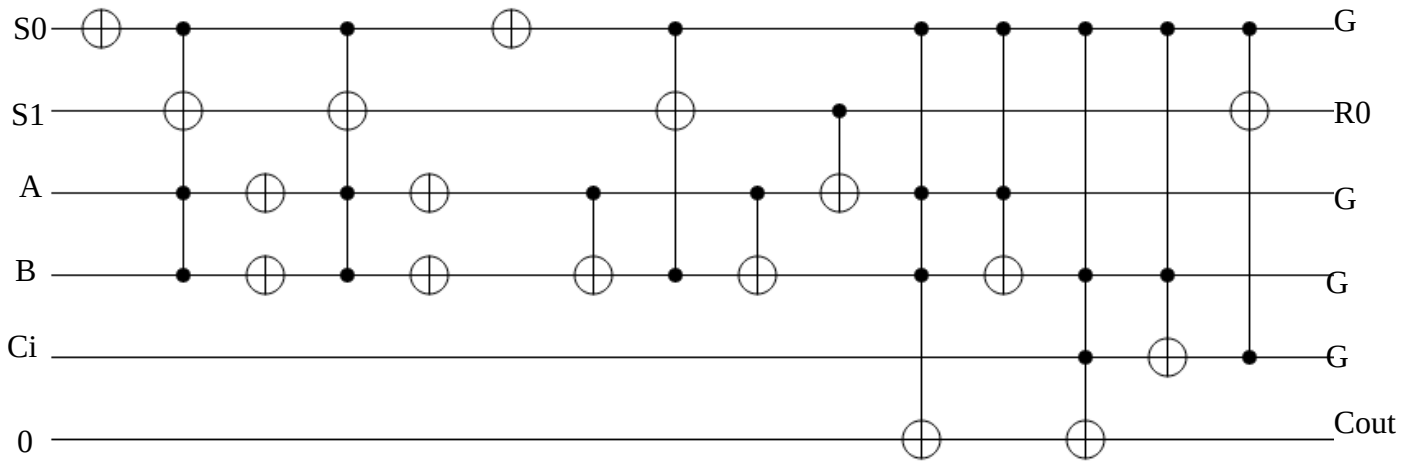


Fig 26 Quantum representation 2 bit ALU using Method II

Next we attempt at enhancing the functionality of the above ALU with increased number of inputs and select input bits as follows:

**ENHANCED FUNCTIONALITY 4 BIT ALU DESIGN USING METHOD II:**

Table 11- Operation table for a 4 bit ALU

INPUTS			OUTPUTS
S1	S2	S3	ARITHMATIC/ LOGICAL OPERATIONS
0	0	0	NOR
0	0	1	OR
0	1	0	ADD
0	1	1	SUBTRACT
1	0	0	XOR
1	0	1	XNOR
1	1	0	AND
1	1	1	NAND

For the above ALU , the select input S3 decides an operation or its inverse is to be performed.

The select inputs S2,S3 decides which logical or arithmetic operation is to be performed.

For eg. If S3 is 0 with S1, S2 =00, an AND operation is performed and its inverse is performed when S3 is 1(NAND).

**The following algorithm has been used for the implementation of an N bit ALU[10]:**

Variables: i,j,N

Input : N,array ai , bi ( $0 \leq i \leq (N - 1)$ ), array ci<sub>j</sub> ( $0 \leq j \leq N$ ), array Sk ( $0 \leq k \leq 2$ ), cin

Output: All the reversible gates to produce Si , ( $0 \leq i \leq (N - 1)$ ), Cout

Initialization: Cout = cin

for i ← 0 to (N - 1) do

if S1S2 = 11 then

Si = XOR(AND(ai , bi ), S3)

end

else if S1S2 = 10 then

Si = XOR(ai , bi , S3 )

end

else if S1S2= 00 then

Si = XOR(AND(NOT (ai ), NOT (bi )), S3 )

end

else if S1S2= 01 then

Si = XOR(ai, bi , ci<sub>i</sub>)

ci<sub>i+1</sub> = XOR(AND(XOR(S3 , ai ), bi ),(AND(XOR(S3 , ai , bi ), ci<sub>i</sub>))

end

end

## **BUILDING BLOCKS FOR A REVERSIBLE ALU USING MODULAR QUANTUM BLOCKS:**

### **1) AND/NAND GATE:**

The foremost module of the proposed ALU architecture is a controlled AND-NAND network, shown in Figure below. According to the table above, this AND-NAND

network gets activated for a control sequence of 11X. Here 'X' could be '0' or '1' depending upon the AND or NAND circuit i.e. if S3 is logic Zero, the AND of a and b is produced otherwise a NAND of a and b is produced.

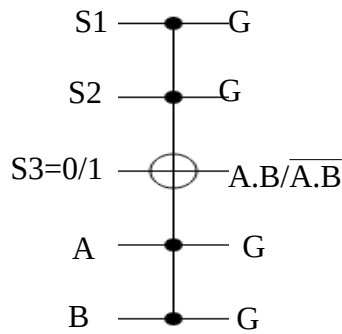


Fig 27 AND/NAND quantum structure

## 2) XOR/XNOR GATE:

The next module is a controlled XOR-XNOR network. According to the table above, this AND-NAND network gets activated for a control sequence of 10X. Here 'X' could be a '0' or '1', i.e. if S3 is low (high) XOR (XNOR) of a and b is performed. The Left hand side figure shows a controlled XOR- XNOR network and the Right Hand Side figure below shows the connection of this XOR/XNOR module in the main ALU network with multiplexing by the select lines.

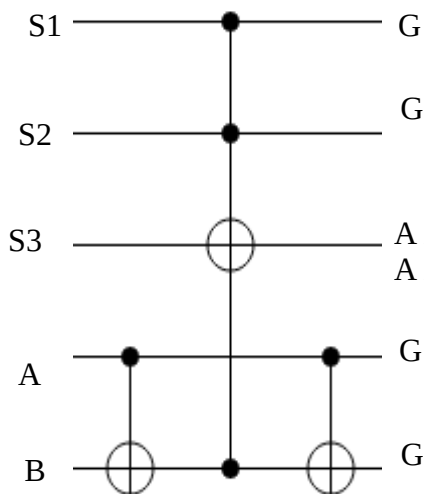


Fig 28 XOR/XNOR quantum structure

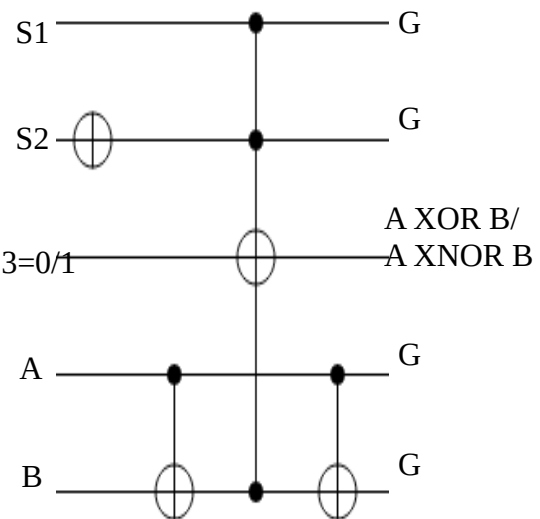


Fig 29 XOR/XNOR quantum structure in the main ALU

### 3) OR/NOR GATE:

The next module is a controlled OR-NOR network. According to the table above, this AND-NAND network gets activated for a control sequence of 00X. Here 'X' could be a '0' or '1', i.e. if S3 is low (high) NOR (OR) of a and b is performed. The Left hand side figure shows controlled OR-NOR network and the Right Hand Side figure below shows the connection of this OR/NOR module in the main ALU network with multiplexing by the select lines.

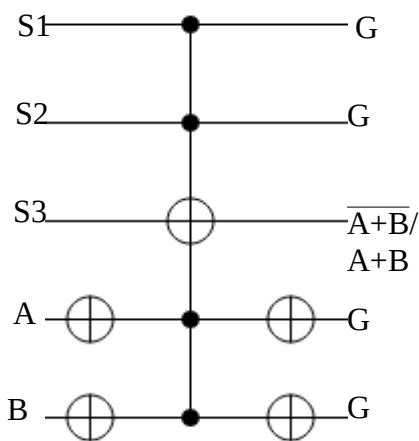


Fig 30 XOR/XNOR quantum structure

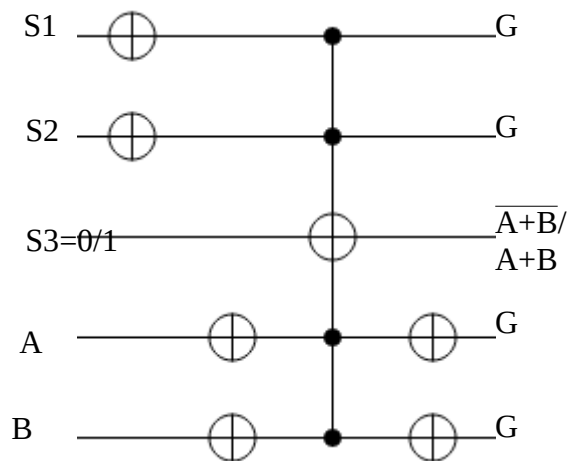


Fig 31 XOR/XNOR quantum structure in the main ALU

### 4) ADDER/SUBTRACTOR:

The fourth and the last module is the Addition-Subtraction network. This module gets activated for a control sequence of 01X. Here 'X' could be a '0' or '1', i.e. if S3 is low (high) ADDITION (SUBTRACTION) of a and b is performed. The adder/Subtractor module is implemented using peres gates as described below.



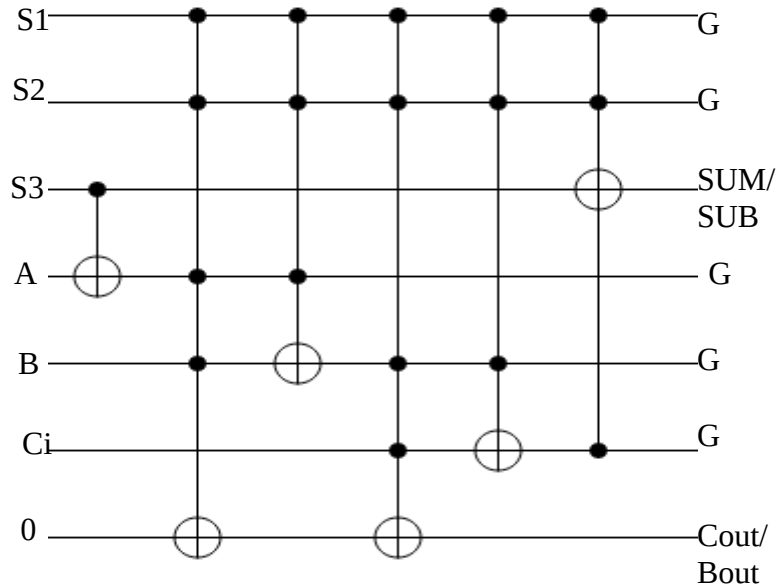


Fig 32 ADDER/SUBTRACTOR  
quantum structure

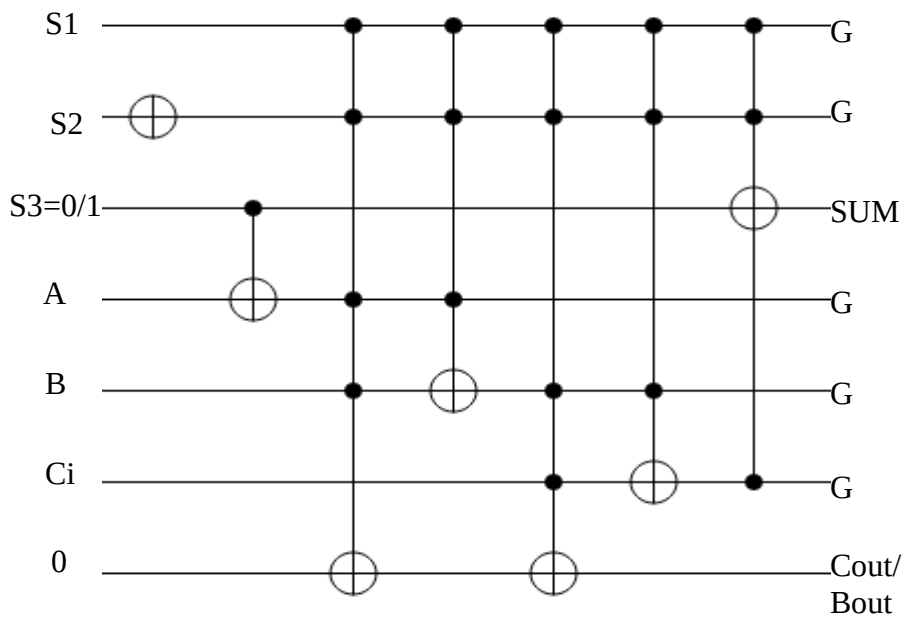


Fig 33 ADDER/SUBTRACTOR  
quantum structure in the main  
ALU

Using the above building blocks for the ALU and multiplexing them with select line signals S0,S1,S2, the design of a 4 bit ALU is accomplished as follows.

An abstract representation of the proposed ALU structure is shown in Fig. 34. The exact reversible ALU structure is shown in Fig. 35 which shows the exact inputs (multibit input bits A and B, select input bits, constant input bits) and exact outputs (garbage output bits - G , arithmetic/logical output bits- R0,R1,R2,Cout/Bout) maintaining a one to one mapping between input and output along with having equal number of input and output bits hence sustaining the necessary conditions of reversibility. It shows the proposed 4 bit ALU with the exact input , output and select bits consisting of a multiplexer to select the desired arithmetic or logical operation to be performed and a function generator to carry out the selected function on the multibit input (A and B).

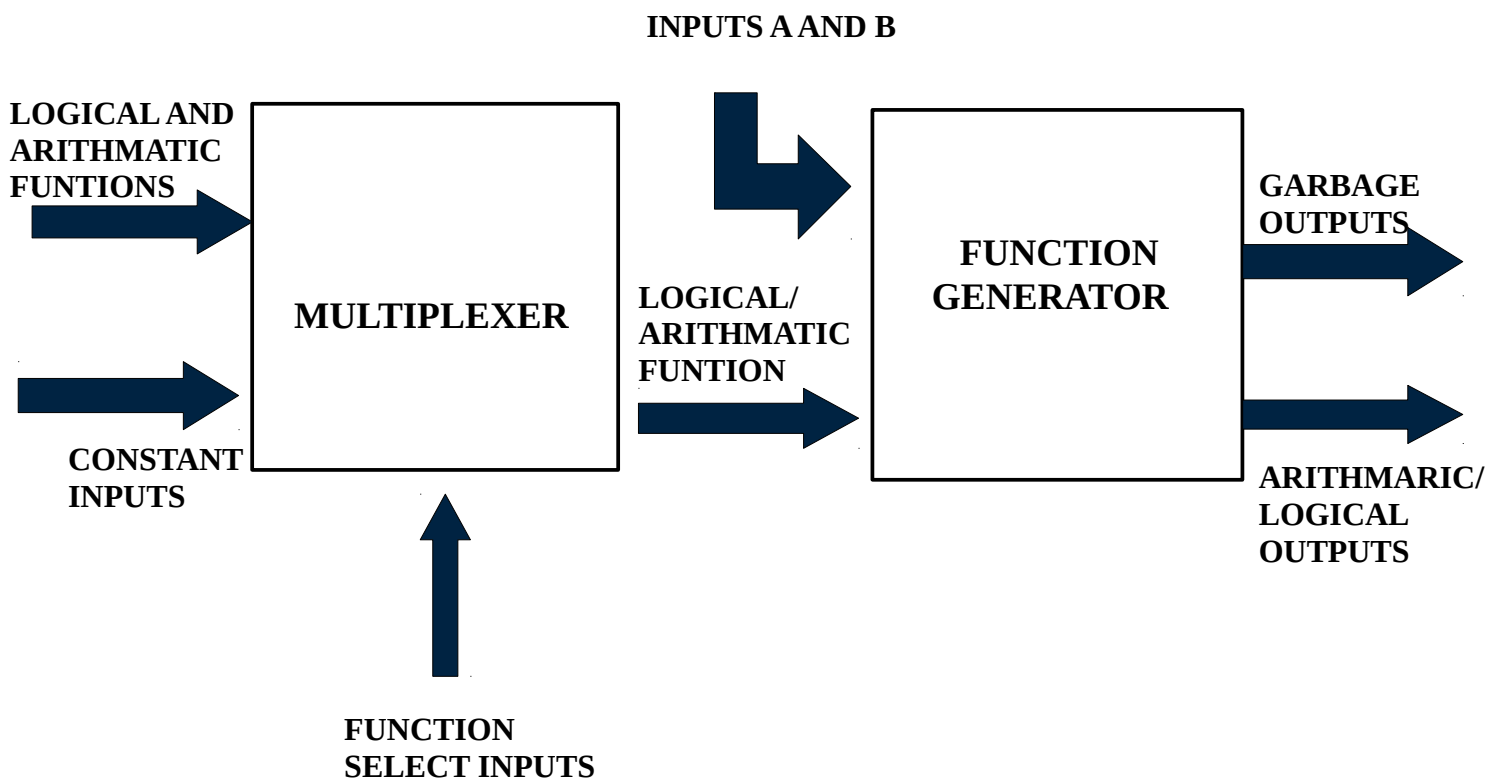


Fig 34 An abstract version of the 4 bit ALU

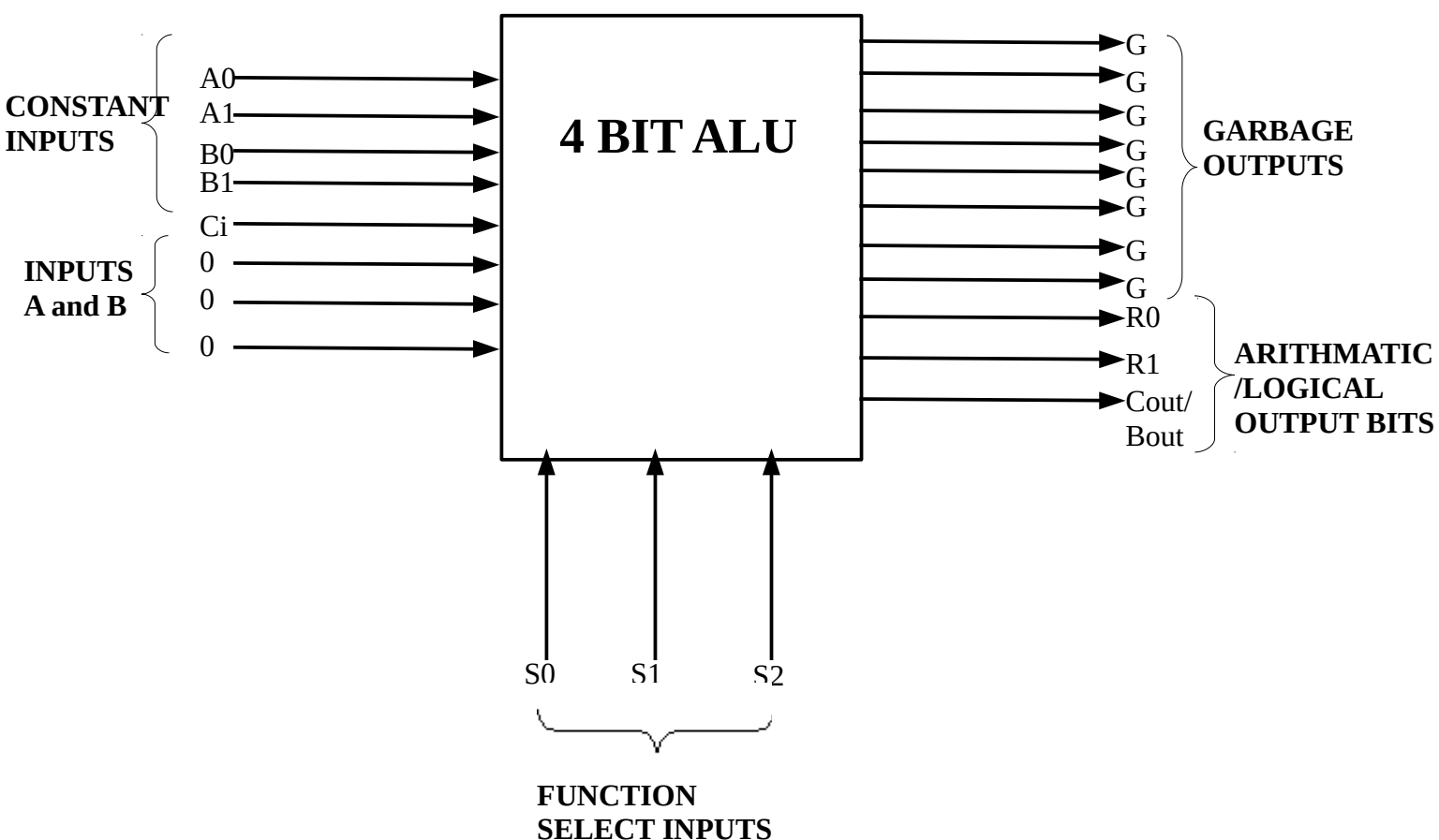


Fig 35 Four bit ALU representation with exact pins

The design has got 3 ancilla input signals(Bit 6,7,8) and 8 garbage outputs(G),3 ALU output bits(Bit 9,10,11).

This ALU implementation is better terms of ancilla inputs and garbage outputs when compared to the previous design for the same.A more detailed comparision of the results is given in section 5.1.4 of the dissertation.

## QUANTUM EQUIVALENT NETWORK FOR 4 BIT ALU:

The following quantum circuit is implemented and simulated on QUIRK(Fig. 36). The outputs bits are signified with R0,R1(Output Bit 0-1) and Cout/Bout. The input bits are the select pins S0,S1,S2 with 2 bit inputs A and B, carry input Ci and 3 ancilla bits .

$$R0 = A0 \diamond B0$$

$$R1 = A1 \diamond B1$$

$$\text{Cout/Bout} = \text{Carry\_output} ( A + B + \text{Cin}) / \text{Borrow\_output}( A - B - \text{Cin})$$

Here," $\diamond$ " represents a logical/arithmetic operation.

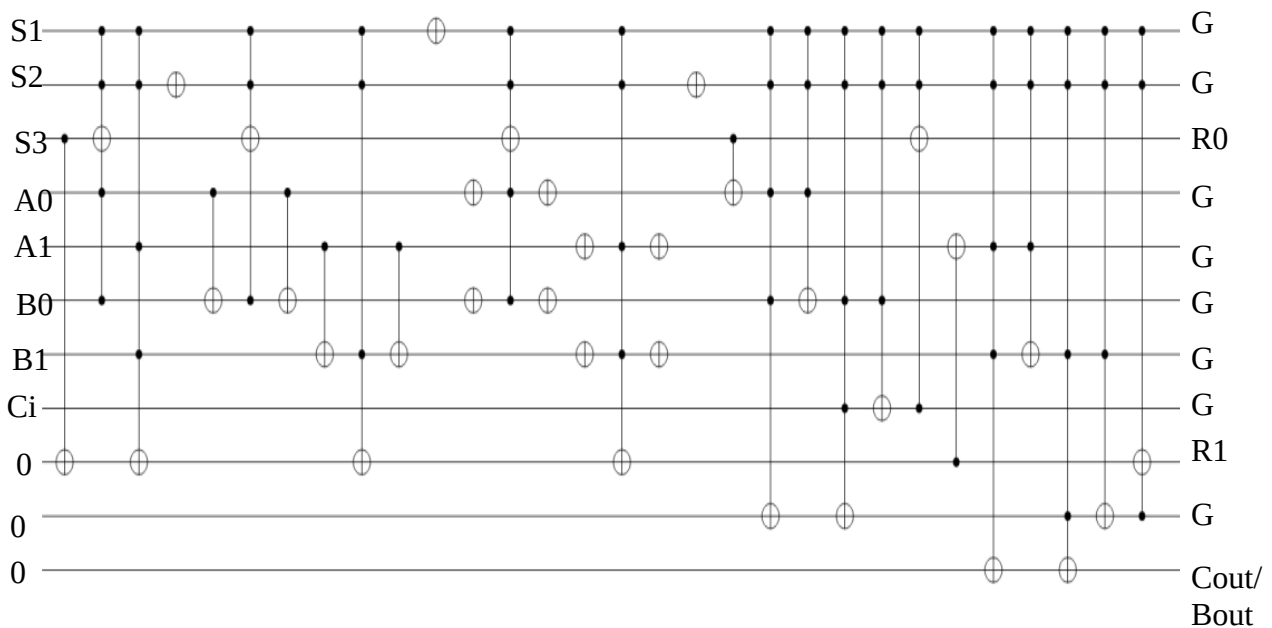


Fig 36 Quantum representation of 4 bit ALU, using Method II

## ENHANCED FUNCTIONALITY 6 BIT ALU DESIGN USING METHOD II :

Using the previously described Method II for the synthesis of an ALU, by combining together the necessary basic building blocks and multiplexing them with select inputs , the following 6 Bit ALU is constructed. An abstract representation of a 6 bit ALU is shown in Fig. 37.

The operation table for the ALU is as follows:

Table 12- Operation table for a 6 bit ALU

INPUTS			OUTPUTS
S1	S2	S3	ARITHMATIC/ LOGICAL OPERATIONS
0	0	0	AND
0	0	1	NAND
0	1	0	XOR
0	1	1	XNOR
1	0	0	NOR
1	0	1	OR
1	1	0	ADD
1	1	1	SUBTRACT

For an N input ALU, the logical operations can be performed independently for each individual bit( for eg. A0-B0,A1-B1,A2-B2) of the multibit input signals A and B , but when it comes to arithmetic operations , there is a dependancy from the previous bit operation. For example , when we perform multibit addition or subtraction, the carry or borrow has to be forwarded from one logical level to the next one to give the final carry and sum output. Fig. 38 shows the proposed 6 bit ALU with the exact input , output and select bits consisting of a multiplexer to select the desired arithmetic or logical operation to be performed and a function generator to carry out the selected funtion on the multibit input (A and B).

Hence to satisfy the abovementioned dependancy of the carry or borrow, the carry/borrow bit has to be copied to be used as input to the next level input combination. This bit copying is done by a fenyman gate ( or a tofolli gate) where in the target bit is set to constant zero bit and the bit to be copied is given as the control bit to it. Also the for each increase in the input signal bitwidth the S3 bit has to be copied every time.

So, By making use of the described Method II of multiplexing the control signals for the ALU and copying the S3,Carry/Borrow bit for each increment in the bitwidth processed per unit of time, the following 6 bit ALU is created. This ALU has a word length of 3 bits each viz. A0,A1,A2 and B0,B1,B2.

The control bit S3 is copied two times(for A1-B1,A2-B2) so as to create the parallel multiplexing for each of the 3 bits per unit time.

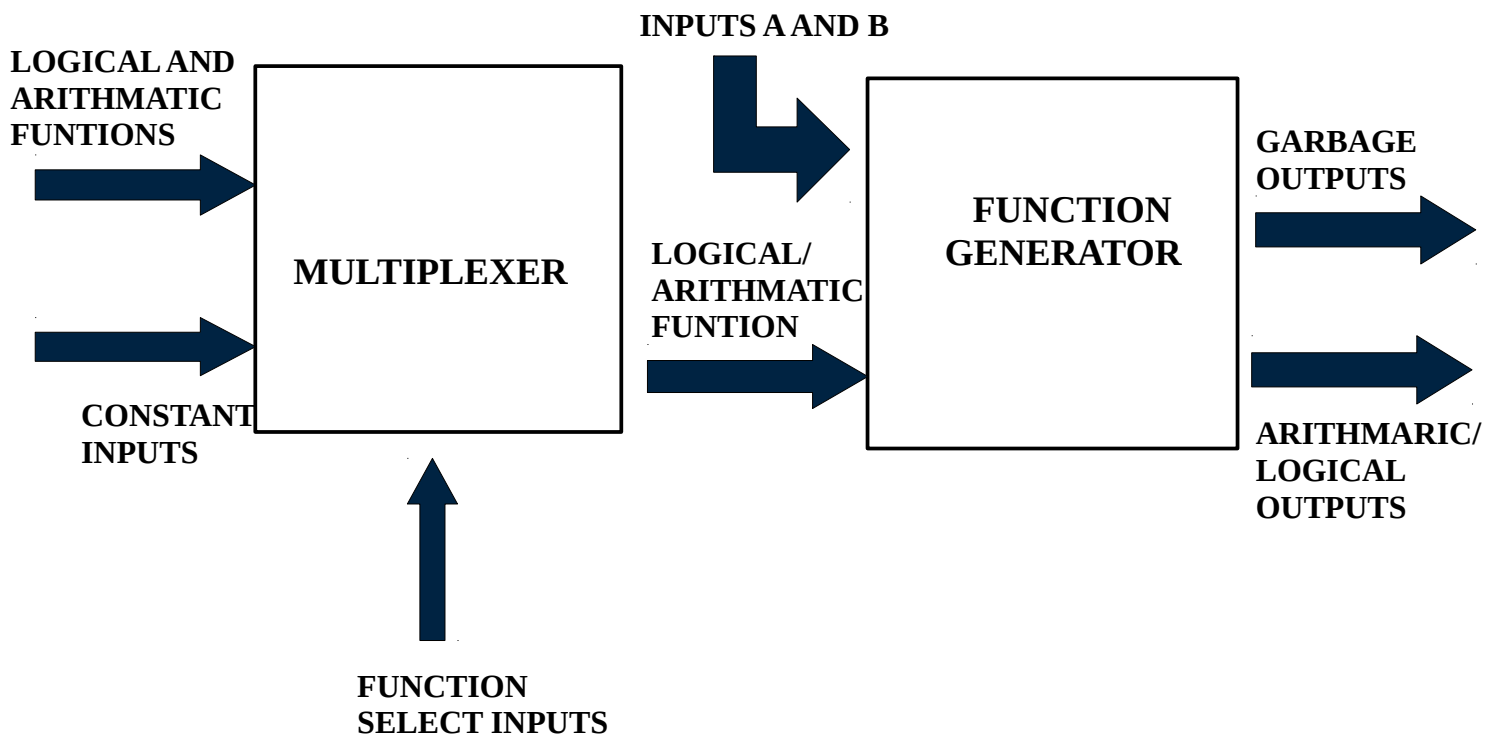


Fig 37 An abstract version of the 6 bit ALU

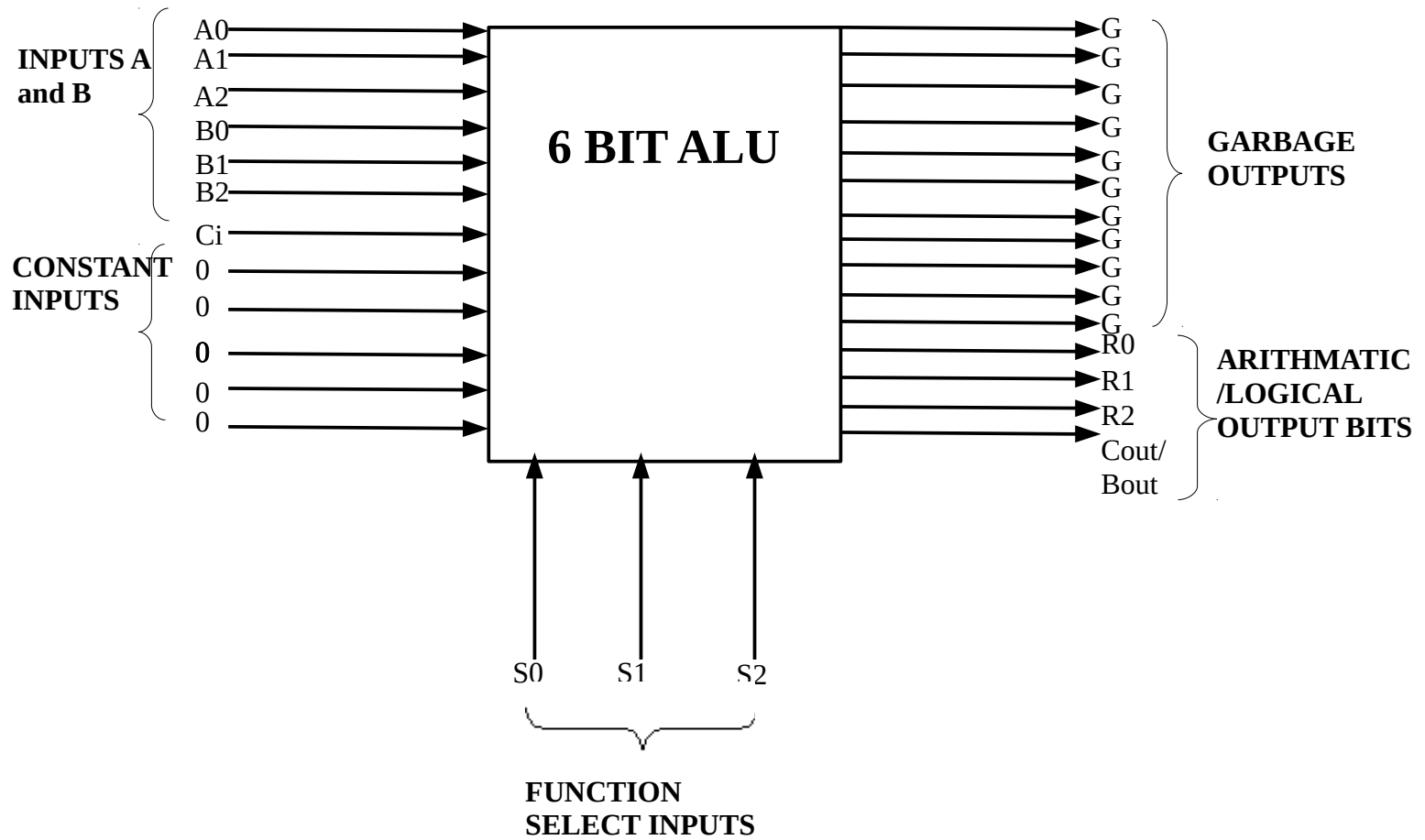


Fig 38 Six bit ALU representation with exact pins

### QUANTUM EQUIVALENT NETWORK FOR 6 BIT ALU:

The following quantum circuit is implemented and simulated on QUIRK(Fig. 39). There are 5 ancilla bits with 11 garbage outputs. The 10<sup>th</sup> constant input bit(ancilla bit) 0, input is used for copying the control bit S3 for bit combination A2-B2 , similarly the 11<sup>th</sup> constant input bit 0, is used for copying the control bit S3 for the input bit combination A2-B2.

$$R0 = A0 \diamond B0$$

$$R1 = A1 \diamond B1$$

$$R2 = A2 \diamond B2$$

$$\text{Cout/Bout} = \text{Carry\_output} ( A + B + \text{Cin}) / \text{Borrow\_output}( A - B - \text{Cin})$$

Here, "♦" represents a logical/arithmetic operation.

The outputs bits are signified with R0,R1,R2(output bits 0,1 and 2) and Cout/Bout. The input bits are the select pins S0,S1,S2 with 3 bit inputs A and B, carry input Ci and 5 ancilla bits .

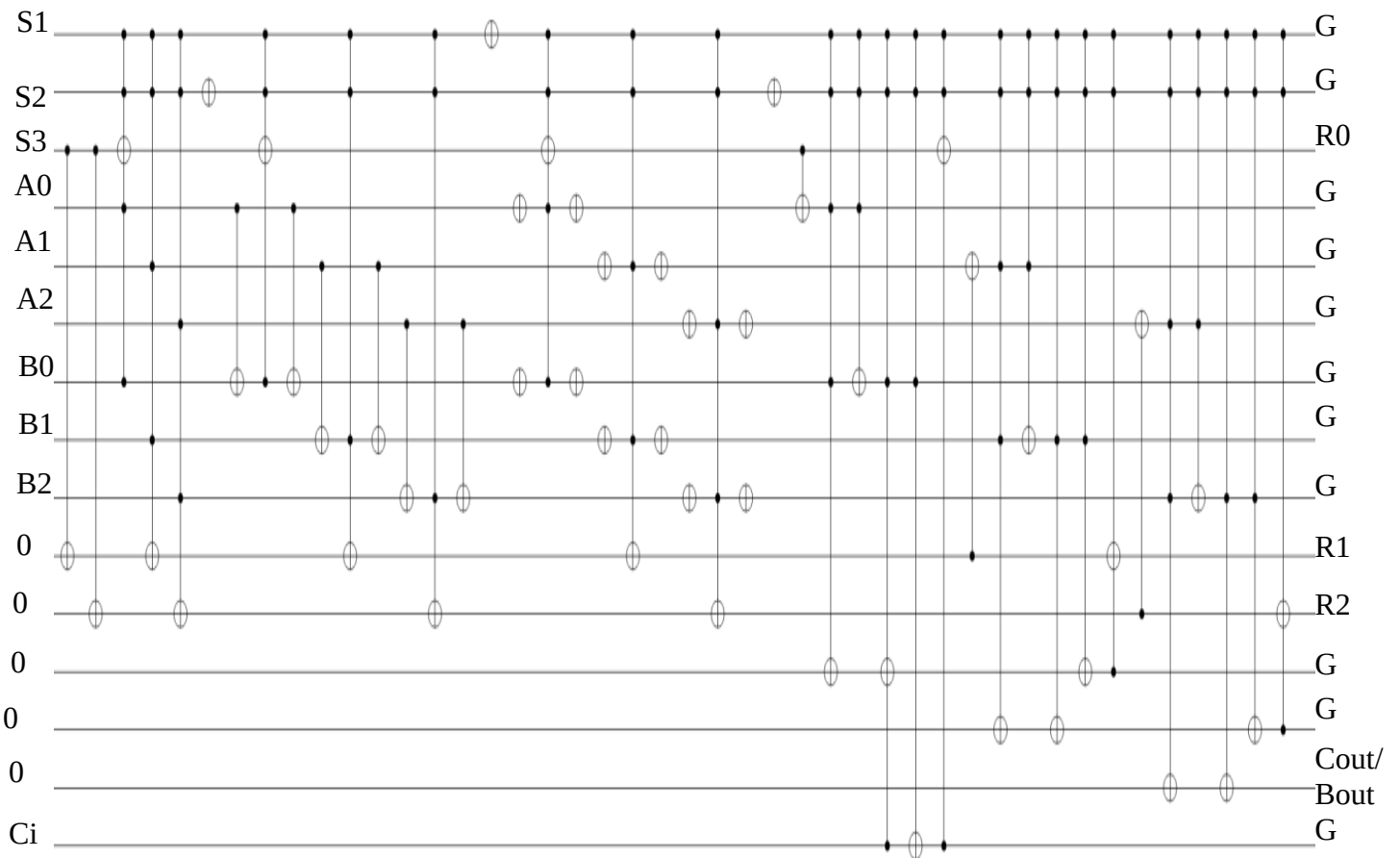


Fig 39 Quantum representation of 6 bit ALU, using Method II



### 5.1.3 CIRCUIT PARAMETERS CALCULATION:

The calculation of various circuit parameters for the Method I is very straight forward and can be calculated directly from the quantum equivalent of the ALU Network by taking into consideration the circuit parameters for the individual reversible gates being used in the Method I ALU synthesis. However so is not the case for the Method II Synthesis approach and have to be calculated as individual quantum module wise. Various circuit parameters calculation for the Method II synthesis approach is summarized as follows.

#### A. Calculation of Ancillary inputs :

For the Method 2, a total of (N-1) copies of S3 are required for output generation and N ancillary input bits are required for carry output generation. Here N is the (number of bits)/2 in an ALU. Thus, the total number of ancillary inputs is given as follows.

$$\text{Ancillary\_inputs}(N) = (N - 1) + N = (2N - 1)$$

#### B. Calculation of Garbage output :

Total number of primary inputs (P.I(N)) ALU network is:

$$\text{P.I}(N) = (3 + 2 \times N + 1) = (2N + 4).$$

Total number of ancillary inputs for the network is:

$$\text{A.I}(N) = (2N - 1).$$

Total number of primary outputs for network is:

$$\text{P.O}(N) = (N+1).$$

Here N is the (number of bits/2) in an ALU.

Therefore total number of garbage outputs is given as:

$$\text{Garbage\_Output}(N) = \text{P.I}(N) + \text{A.I}(N) - \text{P.O}(N) = 2N + 4 + 2N - 1 - N - 1 = (3N + 2)$$

#### C. Calculation of Quantum cost :

Quantum cost for the above Method 2 is calculated by dividing the complete circuit, module wise.

The initial module is an AND – NAND and a module XOR–XNOR is the next one.

The third module is an OR-NOR module, followed by an ADD – SUBTRACT module.

For the AND –NAND module, the quantum cost is

$(6N + 3)$ .

For the XOR – XNOR module, the quantum cost is

$(5N + 2)$ .

For the OR – NOR module, the quantum cost is

$(9N + 5)$ .

The last module being the ADD – SUBTRACT module whose quantum cost is

$(15N + 7)$ .

Here N is the (number of bits/2) in an ALU.

Hence the total quantum cost is given as follows:

$$\begin{aligned}\text{Quantum\_Cost}(N) &= (6N + 3) + (5N + 2) + (9N + 5) + (15N + 7) \\ &= (35N + 17)\end{aligned}$$

#### **D. Calculation of Delay :**

For delay calculation, we take a unit delay as a delay for any given gate. Therefore, the total gate count should be the propagation delay.

For instance, we see that two Toffoli gates acts in parallel for the AND-NAND module. Thus, the module's complete delay is taken as a unity delay.

This implies that net delay in propagation is the complete number of gates operated sequentially.

For a 4 bit ALU, it can be seen that in the proposed ALU architecture there are a total of 14 parallel stages acting sequentially and total N sequential stages occurring only in the ADDITION/SUBTRACTION module.

In this way it is possible to find the delay for any arbitrary bit length ALU, thus expressing the total propagation delay for an N bit ALU as follows:

$$\text{Delay}(N) = (N + 14)$$

### 5.1.4 RESULT AND ANALYSIS:

Table 13 gives a comparison among the different types of ALU. The number of ancilla bits and garbage output is lesser when compared to the existing designs for a 2 bit ALU.

The result below is for comparison between the 2 bit ALU implemented using both Method I and Method II:

Table 13 Results for 2 Bit ALU using Method I and Method II

SYNTHESIS APPROACH	ALU	GARBAGE OUTPUT	ANCILLA BITS	DELAY
EXISTING DESIGN [13]	2 BIT	9	5	26
METHOD I	2 BIT	9	5	19
METHOD II	2 BIT	4	1	15

Method II proves to be much better than Method I in terms of all the three important quantum circuit parameters viz. Garbage outputs, ancilla bits and delay.

To further emphasize on the Method II synthesis approach's efficiency, the result for synthesis of an enhanced functionality ALU using Method II is tabulated as follows:

Table 14 Enhancements in ALU functionality using Method II

SYNTHESIS APPROACH	ALU	GARBAGE OUTPUT	ANCILLA BITS	DELAY
METHOD II	4 BIT	8	3	16
	6 BIT	11	5	17

An ALU is a significant microprocessor element and a central processing unit element. For every computer, an ALU acts as a heart to its instruction execution portion. Taking into consideration the above mentioned results it can be successfully stated that using Method II, an N bit ALU can be formulated and implemented very efficiently with the best possible quantum circuit parameters (garbage outputs, ancilla bits and delay).

## CHAPTER 6- CONCLUSION AND FUTURE WORK

### 6.1 CONCLUSION:

The primary goal of this work was to design an Arithmetic and Logic Unit (ALU) using reversible logic gates. The aim of this project was to suggest a basic computing block, built using reversible gates that will play a crucial part in the evolving creative computing. Methods such as quantum computing. ALU is certainly a needed construction block for a computer's central processing unit (CPU), but has more overhead computing than others in the irreversible domain. However this overhead is an active area of research to be reduced to a minimum because along with the overheads, reversible computing comes across with a huge advantage of low computation power hence paving ways to the future of low power designs.

To create the quantum equivalents of the basic reversible gates used in construction of the building blocks of the ALU, the Bidirectional synthesis algorithm has been used. Two methods for synthesis of a reversible ALU for programmable computing devices has been described.

The first method uses a multiplexer device to synthesize an ALU as well as control signals to execute four different and important arithmetic and logical operations i.e. AND, EXOR, OR and addition. Nevertheless, for such tiny features, the amount of ancilla inputs, trash outputs and quantum price is too big.

To cut on these resources(ancilla inputs, garbage outputs and quantum cost), a new synthesis method is made into use.

On the other side, the second method makes use of structural modularity and operational flexibility to build an ALU using modular quantum structures and multiplex them one by one using selected lines.

Experimental(Simulation) results suggests that realization of reversible circuit using the second method proves to be a better one in terms of ancilla input bits,garbage output bits and quantum cost. Using the method 2, a multibit ALU which processes N bits at a time can be constructed for various reversible computation and processing devices. This proposed design of reversible ALU with Method II is low on grounds of the number of quantum gates used, garbage outputs and quantum Cost and hence can be efficiently used for low power applications.

## 6.2 FUTURE WORK

As such currently, there are still plenty of challenges before reversible logic can be actually thought of turning into a practical competitive technology. A major limitation on today's CMOS technology is represented by the increased heat dissipation per silicon chip. Therefore, reversible circuits presents a very promising low power dissipation computation technology. Reversible logic synthesis, in addition, also has a very close relationship with quantum logic synthesis, so reversible logic synthesis methods can be used to introduce even more effective quantum logic.

This is why the study of reversible logic synthesis is an active area of studies as it will contribute to advancement in associated areas of studies such as ultra-low power IC design and quantum computing. This dissertation work focuses primarily on circuit design based on reversible logic. Proposed work is shown to be effective in terms of trash outputs, continuous inputs, complexity of logic, use of devices, power consumption and delay.

Reversible logic plays a key role in the progress of associated areas of studies such as low-power design and high-speed devices. An intriguing future work in this direction could be to develop an efficient complete reversible computer architecture with the help of the suggested(proposed) ALU design. Reversible logic offers big performance advantages over the present conventional technology, but its still has a long and expensive technology development path to replace the existing conventional technologies from the market.

### 6.3 REFERENCES:

1. R. Landauer, "Irreversibility and Heat Generation in the computational process", IBM journal of Research and Development, Vol. 05,1961
2. C.H. Bennet, "Notes on the History of Reversible Computation", IBM Journal of Research and Development, Vol. 32, 1998.
- 3.P.K LALA, J.P.PARKERSON, P.CHAKRABORTY Electrical Engineering Department, Texas A&M University-Texarkana, Texas 75503, USA Computer Science and Computer Engineering Department University of Arkansas, Fayetteville, Arkansas, 72701, "Adder Designs using Reversible Logic Gates", USA WESAS TRANSACTIONS on CUIRCUITS AND SYSTEMS, Issue 6, Volume 9, June 2010.
4. Tiny Hey. *Quantum Computing: An Introduction. Computing and Control Engineering Journal*, 10(3): 105-112, 1999
- 5.Michael A Nielsen and Isaac L Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2010.
- 6.IBMQExperienceDocumentation, [https://quantumexperience.ng.bluemix.net/proxy/tutorial/full-user-guide\\_introduction.html](https://quantumexperience.ng.bluemix.net/proxy/tutorial/full-user-guide_introduction.html)
- 7.Quantum Entanglement From Wikipedia, the free encyclopedia-[https://en.wikipedia.org/wiki/Quantum\\_entanglement](https://en.wikipedia.org/wiki/Quantum_entanglement)
8. Shaveta Thakral, Dipali Bansal, S.K. Chakarvarti, "Review of Truth Table Based Reversible Logic Synthesis Methods", 2015 International Conference on Soft Computing Techniques and Implementations- (ICSCTI) Department of ECE, FET, MRIU, Faridabad, India, Oct 8-10, 2015.
9. Arindam Banerjee, Debesh Kumar Das, "A New ALU Architecture Design using Reversible Logic", Sixth International Symposium on Embedded Computing and System Design (ISED), 2016.
10. Y. Syamala , A. V. N. Tilak, "Reversible Arithmetic Logic Unit", IEEE 3rd International Conference on Electronics Computer Technology , 2011.
11. Bra-ket notation, From Wikipedia, the free encyclopedia, [https://en.wikipedia.org/wiki/Bra%E2%80%93ket\\_notation](https://en.wikipedia.org/wiki/Bra%E2%80%93ket_notation)
12. QUIRK , how to use quirk,

<https://github.com/Strilanc/Quirk/wiki/How-to-use-Quirk>

13. Deeptha A, Drishika Muthanna, Dhrithi M, Pratiksha M, B S Kariyappa , “Design and Optimization of 8 bit ALU using Reversible Logic”, IEEE International Conference On Recent Trends In Electronics Information Communication Technology, May 20-21, 2016,

14. The bloch sphere , from Wikipedia, the free encyclopedia,  
[https://en.wikipedia.org/wiki/Bloch\\_sphere](https://en.wikipedia.org/wiki/Bloch_sphere)

15. Quantum superposition, from Wikipedia, the free encyclopedia,  
[https://en.wikipedia.org/wiki/Quantum\\_superposition](https://en.wikipedia.org/wiki/Quantum_superposition)

16. Mathias Soeken , Stefan Frehse, Robert Wille , and Rolf Drechsler , “Institute of Computer Science, University of Bremen, 28359 Bremen, Germany Cyber-Physical Systems, DFKI GmbH, 28359 Bremen, Germany”, De Vos A., Wille R. (eds) Reversible Computation. RC 2011. Lecture Notes in Computer Science, vol 7165. Springer, Berlin, Heidelberg, 2012.

17. Grover’s Algorithm, From Wikipedia, the free encyclopedia,  
[https://en.wikipedia.org/wiki/Grover%27s\\_algorithm](https://en.wikipedia.org/wiki/Grover%27s_algorithm)

# Thesis

---

## ORIGINALITY REPORT

---

<b>16%</b>	<b>5%</b>	<b>8%</b>	<b>10%</b>
SIMILARITY INDEX	INTERNET SOURCES	PUBLICAT IONS	STUDENT PAPERS

---

## PRIMARY SOURCES

---

<b>1</b>	<b>Submitted to American University of Athens</b> Student Paper	<b>3%</b>
<b>2</b>	<b>Arindam Banerjee, Debesh Kumar Das. "A new ALU architecture design using reversible logic", 2016 Sixth International Symposium on Embedded Computing and System Design (ISED), 2016</b> Publicat ion	<b>3%</b>
<b>3</b>	<b>quantum.phys.cmu.edu</b> Internet Source	<b>1%</b>
<b>4</b>	<b>Submitted to Yeshwant Rao Chavan College of Engineering</b> Student Paper	<b>1%</b>
<b>5</b>	<b>edoc.pub</b> Internet Source	<b>1%</b>
<b>6</b>	<b>ijmece.org</b> Internet Source	<b>&lt;1%</b>
<b>7</b>	<b>Submitted to Visvesvaraya Technological University</b>	<b>&lt;1%</b>

---



8

"Transactions on Computational Science XXIV",  
Springer Science and Business Media LLC,  
2014

Publication

<1%

---

9

Submitted to Symbiosis International University

Student Paper

<1%

---

10

[www.slideshare.net](http://www.slideshare.net)

Internet Source

<1%

---

11

Submitted to The University of the South  
Pacific

Student Paper

<1%

---

12

Submitted to Lovely Professional University

Student Paper

<1%

---

13

[www.researchgate.net](http://www.researchgate.net)

Internet Source

<1%

---

14

Chinmay Sharma, Hitesh Pahuja, Mandeep  
Dadhwal, Balwinder Singh. "Study of  
Reversible Logic Synthesis with Application in  
SOC: A Review", IOP Conference Series:  
Materials Science and Engineering, 2017

Publication

<1%

---

15

Submitted to University College London

Student Paper

<1%

---

16

Submitted to SASTRA University

Student Paper

<1 %

17

Rakhi Saha, Sambita Dalal. "A novel reversible combinational circuit design for low power computation", 2015 IEEE Power, Communication and Information Technology Conference (PCITC), 2015

Publication

<1 %

18

[theaccents.org](http://theaccents.org)

Internet Source

<1 %

19

Submitted to Kingston College

Student Paper

<1 %

20

[ijcta.com](http://ijcta.com)

Internet Source

<1 %

21

"Reversible Computation", Springer Science and Business Media LLC, 2017

Publication

<1 %

22

Submitted to University of Southampton

Student Paper

<1 %

23

[www.engpaper.net](http://www.engpaper.net)

Internet Source

<1 %

24

Submitted to Universiti Teknologi MARA

Student Paper

<1 %

25

Submitted to Indian Institute of Technology, Madras

<1 %

