

# A Study on Reversible Logic Gates of Quantum Computing

A.G.Aruna<sup>1</sup>, K H Vani<sup>2</sup>, C Sathya<sup>3</sup>, R.Sowndarya Meena<sup>4</sup>

<sup>1,2,3</sup>Assistant Professor, Department of Computing,  
Coimbatore Institute of Technology,  
Coimbatore, Tamil Nadu, India.

<sup>4</sup>Student, Department of MCA,  
Coimbatore Institute of Technology,  
Coimbatore, Tamil Nadu, India.

**Abstract - Circuit model of computer provides a better abstraction of computational process across the world. The building block of a circuit is the logic gates which are the main information processing units. In electronics, a logic gate is an idealized or physical device implementing a Boolean function; that is, it performs a logical operation on one or more logical inputs, and produces a single logical output. Similar to classical computer, logic gates play an important role in building circuits in a quantum computer. In this paper, a study of logic gates and the various mathematical representations of logic gates has been done.**

**Keywords:** *Quantum Logic Gate; Quantum Computing; Elementary gates; Logic gates;*

## I. INTRODUCTION

In the circuit model, computer scientists regard any computation as being equivalent to the action of a circuit built out of a handful of different types of Boolean logic gates acting on some binary (i.e., bit string) input. Each logic gate transforms its input bits into one or more output bits in some deterministic fashion according to the definition of the gate.

The classical computation is being performed using Boolean logic which uses variables 0 and 1[1].The physical interpretation of 0 and 1 is the voltage On/Off. But in case of quantum computation, quantum mechanics provides a new set of rules that go beyond this classical paradigm. The basic variable in quantum computing [2, 3, 4] is a quantum bit which is represented as a vector in a two dimensional complex Hilbert space. Suppose the levels, or eigenstates of the quantum variable, are labeled  $|0\rangle$  and  $|1\rangle$ . This has a direct correspondence with the discrete states of a classical bit, 0 and 1. However, a qubit is a quantum state, and as such can be in a superposition state also. That is, in addition to  $|0\rangle$  and  $|1\rangle$ , a qubit can exist more generally in the state,  $c_0|0\rangle + c_1|1\rangle$ , where  $c_0$  and  $c_1$  are complex coefficients normalized to 1. This is the main distinction between classical and quantum memory, in that even

though a qubit has discrete eigenstates, there is something analog about it also in the continuous range of superposition states that it can take on.

In recent years quantum computing becomes a forefront research in the field of algorithms, cryptography and artificial intelligence [5,6].We can describe qubits as mathematical objects with certain specific properties. The beauty of treating qubits as mathematical objects is that it gives us the freedom to construct a general theory of quantum computation, which does not depend upon a specific system for its realization. The task of information processing can be performed using quantum mechanical principles. And when quantum principles are applied on information processing it gives the concept of quantum computing.

In quantum mechanics the state of a physical system is represented by its wave function which contains all information to describe the state of the system completely. Contrary to classical computing we carry out computations using quantum states which follow properties of Quantum mechanics. Changes occurring to a quantum state can be explained using the language of quantum computation. As classical computer is built from an electrical circuit containing wires and logic gates, a quantum computer is built from a quantum circuit containing wires and elementary quantum gates to carry out and manipulate the quantum information. This paper addresses the basic properties of classical and quantum gates and a comparison is made between them with emphasizing the shortcomings of classical gates.

## II. LOGIC GATES FOR CLASSICAL COMPUTER

Classical computation theory became prominent after Church and Turing made their investigation into the characteristics of computability in the year 1936 [7]. Logic gates and logic circuits took part major role in the theory of

computation. In the course of time the implementation, sophistication and optimal structure of classical logic circuits have been developed [8].

A. Irreversible classical logic gates

A gate is said to be logically reversible if we can uniquely determine the input values from the output values otherwise the gate is said to be logically irreversible. We will begin our study of classical logic gates by introducing the notion of Boolean functions.

AND-GATE: It is defined as a Boolean Function,

$$f(x, y) = xy \tag{1}$$

The result can also be written as

$$f(x, y) = \begin{cases} 1, & \text{if } x = y = 1 \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

(Meaning: product of x and y)

OR-GATE: It is defined as a Boolean Function,

$$f(x, y) = x + y \tag{3}$$

The result can also be written as

$$f(x, y) = \begin{cases} 0, & \text{if } x = y = 0 \\ 1, & \text{otherwise} \end{cases} \tag{4}$$

(Meaning: plus of x and y)

XOR-GATE : It is defined as a Boolean Function,

$$f(x, y) = x \oplus y \tag{5}$$

The result can also be written as

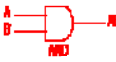
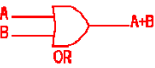


$$f(x, y) = \begin{cases} 0, & \text{if } x = y = 0 \text{ or } x = y = 1 \\ 1, & \text{otherwise} \end{cases} \tag{6}$$

NAND and NOR are universal irreversible logic gates from which the Boolean function can be derived as:

$$\text{NAND: } f(x, y) = \overline{xy} \tag{7}$$

$$\text{NOR: } f(x, y) = \overline{x + y} \tag{8}$$

TABLE I.LOGIC GATES OF CLASSICAL COMPUTER

Logic gate	Symbol	Boolean function	Table															
AND		F=AB	<table border="1"> <caption>2 Input AND gate</caption> <thead> <tr> <th>A</th> <th>B</th> <th>A·B</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> </tr> </tbody> </table>	A	B	A·B	0	0	0	0	1	0	1	0	0	1	1	1
A	B	A·B																
0	0	0																
0	1	0																
1	0	0																
1	1	1																
OR		F=A+B	<table border="1"> <caption>2 Input OR gate</caption> <thead> <tr> <th>A</th> <th>B</th> <th>A+B</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> </tr> </tbody> </table>	A	B	A+B	0	0	0	0	1	1	1	0	1	1	1	1
A	B	A+B																
0	0	0																
0	1	1																
1	0	1																
1	1	1																
XOR		F=A⊕B	<table border="1"> <caption>2 Input EXOR gate</caption> <thead> <tr> <th>A</th> <th>B</th> <th>A⊕B</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> </tr> </tbody> </table>	A	B	A⊕B	0	0	0	0	1	1	1	0	1	1	1	0
A	B	A⊕B																
0	0	0																
0	1	1																
1	0	1																
1	1	0																
NOT		F= A'	<table border="1"> <caption>NOT gate</caption> <thead> <tr> <th>A</th> <th>A'</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> </tr> </tbody> </table>	A	A'	0	1	1	0									
A	A'																	
0	1																	
1	0																	

III. LIMITATIONS OF IRREVERSIBLE LOGIC GATES

The first concerns about the reversibility of computation were raised in the 1970s. There were two related issues, logical reversibility and physical reversibility, which were intimately connected. Logical reversibility refers to the ability to reconstruct the input from the output of a computation, or gate function. For instance, the NAND gate is explicitly irreversible, taking two inputs to one output, while the NOT gate is reversible (it is its' own inverse). The connection to physical reversibility is usually made as follows. Since the NAND gate has only one output, one of its' inputs has effectively been erased in the process, whose information has been irretrievably lost. The change in entropy that we would associate with the lost of one bit of information is ln 2, which, thermodynamically, corresponds to an energy increase of kT ln 2, where k is Boltzman's constant and T is the temperature. The heat dissipated during a process is usually taken to be a sign of physical irreversibility, that the microscopic physical state of the system cannot be restored exactly as it was before the process took place.

In the 70s, there were two questions, one was whether a computation can be done in a logically reversible fashion (unlike one that uses NAND gates, for example), and the other was whether any heat needs to be dissipated during a computation. Both of these issues were quite academic however, since as Feynman pointed out [6], an actual

transistor dissipates close to  $10^{10}$  kT of heat, and even the DNA copying mechanism in a human cell dissipates about 100 kT of heat per bit copied (which is understandable from a consideration of the chemical bonds that need to be broken in the process), both far from the ideal limit of  $kT \ln 2$  for irreversible computing.

That classical computation can be done reversibly with no energy dissipated per computational step was discovered by Bennett in 1973 [3]. He showed this by constructing a reversible model of the Turing machine – a symbolic model for computation introduced by Turing in 1936 [1] – 5 and showing that any problem that can be simulated on the original irreversible machine can also be simulated with the same efficiency on the reversible model. The logical reversibility inherent in the reversible model implied that an implementation of such a machine would also be physically reversible. This started the search for physical models for reversible classical computation, a review of which is given in [5].

The model of reversible computation has the number of inputs and outputs of the function  $f$  will be the same, and  $f$  will be required to be a one-to-one Boolean function. Likewise, we can pose the problem of universality as before, and ask for a set of universal reversible logic gates that can simulate arbitrary reversible Boolean functions. Since reversible logic gates are symmetric with respect to the number of inputs and outputs, we can represent them in ways other than the truth table, that emphasizes this symmetry. We have already encountered the reversible NOT gate, whose truth table was given in tables (1). We could also write this in the form of a matrix, or as a graphic, gate. Again, as shown in Fig. 5, the CN gate can be shown to be manifestly reversible by putting two CN gates back to back.

A	NOT A
0	1
1	0

Fig. 1 Truth Table for NOT gate

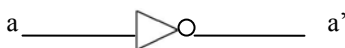


Fig. 2 Alternative symbols for NOT gate.

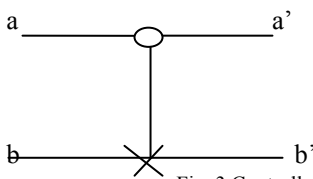


Fig. 3 Controlled NOT or CN gate.

Any logical operation can be built from one of several complete sets of classical logic gates - a choice from NOT, AND, OR, XOR, NAND and so on. Similarly, one can show that there are complete sets of reversible gates that allow us to perform any logic operation. In fact, we need more than just the CN gate: we can add a Controlled Controlled NOT (CCN) or ‘Toffoli’ gate (Fig. 6) or a more complicated Fredkin exchange gate (Fig. 7).

Why do we care about all this? Well for one thing it is possible that use of such gates may one day be needed to reduce power consumption of microprocessors implemented in CMOS silicon technology. At present, the Intel Pentium discards something like 100,000 bits per flop with each discarded bit incurring at least the minimum Landauer energy loss [11]. In our case, however, we are interested because the laws of quantum physics are reversible in time. This guarantees that probability is conserved as a state evolves with time. Technically speaking, the Schroedinger time evolution operator is unitary and preserves the norm of quantum mechanical states (see below). To build a quantum computer with quantum states evolving according to the Schroedinger equation therefore necessarily requires us to use realisations of reversible logic gates.

a	b	a'	b'
0	0	0	0
0	1	0	1
1	0	1	1
1	1	1	0

Fig. 4 Truth Table for Controlled NOT gate.

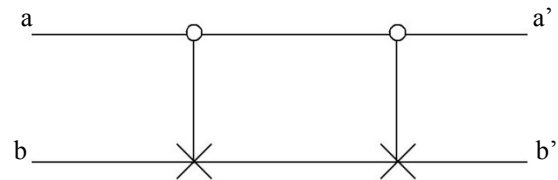


Fig. 5 CN gates are reversible.

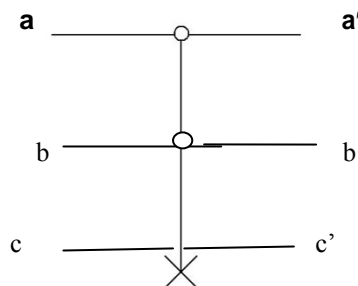


Fig. 6 Controlled Controlled NOT, CCN or Toffoli gate.

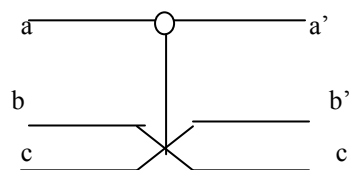


Fig. 7 Fredkin Exchange gate.

#### IV. QUANTUM LOGIC GATES

In quantum computing and specifically the quantum circuit model of computation, a quantum gate (or quantum logic gate) is a basic quantum circuit operating on a small number of qubits. They are the building blocks of quantum circuits, like classical logic gates are for conventional digital circuits.

Unlike many classical logic gates, quantum logic gates are reversible. However, it is possible to perform classical computing using only reversible gates. For example, the reversible Toffoli gate can implement all Boolean functions. This gate has a direct quantum equivalent, showing that quantum circuits can perform all operations performed by classical circuits.

Quantum logic gates are represented by unitary matrices. The most common quantum gates operate on spaces of one or two qubits, just like the common classical logic gates operate on one or two bits. This means that as matrices, quantum gates can be described by  $2 \times 2$  or  $4 \times 4$  unitary matrices.

Quantum gates are usually represented as matrices. A gate which acts on  $k$  qubits is represented by a  $2^k \times 2^k$  unitary matrix. The number of qubits in the input and output of the gate have to be equal. The action of the quantum gate is found by multiplying the matrix representing the gate with the vector which represents the quantum state. In the following, the vector representation of a single qubit is:

$$v_0|0\rangle + v_1|1\rangle \rightarrow \begin{bmatrix} v_0 \\ v_1 \end{bmatrix} \tag{9}$$

and the vector representation of two qubits is:

$$v_{00}|00\rangle + v_{01}|01\rangle + v_{10}|10\rangle + v_{11}|11\rangle \rightarrow \begin{bmatrix} v_{00} \\ v_{01} \\ v_{10} \\ v_{11} \end{bmatrix} \tag{10}$$

Where  $|ab\rangle$  is the state where the first qubit has value  $a$  and the second qubit  $b$ .

##### A. Hadamard gate

The Hadamard gate acts on a single qubit. It maps the basis state  $|0\rangle$  to  $\frac{|0\rangle+|1\rangle}{\sqrt{2}}$  and  $|1\rangle$  to  $\frac{|0\rangle-|1\rangle}{\sqrt{2}}$  represents a rotation of about the axis  $\pi$ . Equivalently, it is the combination of two rotations,  $\pi/2$  about the Y-axis followed by  $\pi$  about the X-axis. It is represented by the

$$\text{Hadamard matrix: } \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \tag{11}$$

Since  $HH^* = I$  where  $I$  is the identity matrix,  $H$  is indeed a unitary matrix. (i.e) In mathematics, a complex square matrix  $U$  is unitary if its conjugate transpose  $U^*$  is also its inverse – that is, if

$$U^*U = UU^* = I \tag{12}$$

where  $I$  is the identity matrix

##### B. Pauli-X gate

The Pauli-X gate acts on a single qubit. It is the quantum equivalent of a NOT gate (with respect to the standard basis  $x |0\rangle, |1\rangle$  which privileges the Z-direction) . It equates to a rotation of the Bloch Sphere around the X-axis by  $\pi$  radians. It maps  $|0\rangle$  to  $|1\rangle$  and  $|1\rangle$  to  $|0\rangle$ . Due to this nature, it is sometimes called bit-flip. It is represented by the

Pauli matrix

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \tag{13}$$

##### C. Pauli-Y gate

The Pauli-Y gate acts on a single qubit. It equates to a rotation around the Y-axis of the Bloch Sphere by  $\pi$  radians. It maps to  $|0\rangle$  to  $i|1\rangle$  and  $|1\rangle$  to  $i|0\rangle$ . It is represented by the Pauli Y matrix:

$$Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} \tag{14}$$

##### D. Pauli-Z gate

The Pauli-Z gate acts on a single qubit. It equates to a rotation around the Z-axis of the Bloch Sphere by  $\pi$  radians. Thus, it is a special case of a phase shift gate (next) with  $\theta=\pi$ . It leaves the basis state  $|0\rangle$  unchanged and

maps to  $|1\rangle$  to  $-|1\rangle$ . Due to this nature, it is sometimes called phase-flip. It is represented by the Pauli Z matrix:

$$Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \tag{15}$$

E. Phase shift gates

This is a family of single-qubit gates that leave the basis state  $|0\rangle$  unchanged and map  $|1\rangle$  to  $e^{i\Phi}|1\rangle$ . The probability of measuring a  $|0\rangle$  or  $|1\rangle$  is unchanged after applying this gate, however it modifies the phase of the quantum state. This is equivalent to tracing a horizontal circle (a line of latitude) on the Bloch Sphere by  $\Phi$  radians.

$$R_{\Phi} = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\Phi} \end{bmatrix} \tag{16}$$

where  $\Phi$  is the phase shift.

F. Swap gate

The swap gate swaps two qubits. With respect to the basis  $|00\rangle, |01\rangle, |10\rangle, |11\rangle$  it is represented by the matrix:

$$SWAP = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{17}$$

G. Square root of Swap gate

The  $\sqrt{SWAP}$  gate performs half-way of a two-qubit swap. It is universal such that any quantum many qubit gate can be constructed from only  $\sqrt{SWAP}$  and single qubit gates.

$$\sqrt{SWAP} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{1}{2}(1+i) & \frac{1}{2}(1-i) & 0 \\ 0 & \frac{1}{2}(1-i) & \frac{1}{2}(1+i) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{18}$$

H. Controlled gates

Controlled gates act on 2 or more qubits, where one or more qubits act as a control for some operation. For example, the controlled NOT gate (or CNOT) acts on 2 qubits, and performs the NOT operation on the second qubit only when the first qubit is  $|1\rangle$  and otherwise leaves it unchanged. It is represented by the matrix

$$CNOT = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{19}$$

More generally if  $U$  is a gate that operates on single qubits with matrix representation

$$U = \begin{bmatrix} x_{00} & x_{01} \\ x_{10} & x_{11} \end{bmatrix} \tag{20}$$

then the *controlled-U gate* is a gate that operates on two qubits in such a way that the first qubit serves as a control. The matrix representing the controlled  $U$  is

$$C(U) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & x_{00} & x_{01} \\ 0 & 0 & x_{10} & x_{11} \end{bmatrix} \tag{21}$$

I. Toffoli gate

The Toffoli gate, also CCNOT gate, is a 3-bit gate, which is universal for classical computation. The quantum Toffoli gate is the same gate, defined for 3 qubits. If the first two bits are in the state  $|1\rangle$ , it applies a Pauli-X on the third bit, else it does nothing. It is an example of a controlled gate. Since it is the quantum analog of a classical gate, it is completely specified by its truth table.

INPUT			OUTPUT		
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

The Matrix representation of T gate is

$$T = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (22)$$

*J. Fredkin gate*

The Fredkin gate (also CSWAP gate) is a 3-bit gate that performs a controlled swap. It is universal for classical computation. As with the Toffoli gate it has the useful property that the numbers of 0s and 1s are conserved throughout, which in the billiard ball model means the same number of balls are output as input.

INPUT			OUTPUT		
C	I <sub>1</sub>	I <sub>2</sub>	C	O <sub>1</sub>	O <sub>2</sub>
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

$$F = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (23)$$

**CONCLUSION**

In this paper an overview of irreversible and reversible logic gates have been tabulated and represented mathematically. Hence using quantum computer we can overcome the irreversibility nature of classical computation and avoid information loss.

**REFERENCE**

- [1] M.Morris Mano, Digital Design Prentice Hall, Third Edition, (2002)
- [2] Mikio Nakahara and Tetsuo Ohmi, Quantum computing, CRC press, (2008)
- [3] Nielsen, M.A. and Chuang, I.L., Quantum computation and quantum information. UK: Cambridge University Press,(2000)
- [4] Sahni, V., Quantum computing. New Delhi: Tata McGraw Hill, (2007)
- [5] Sitakanta Nayak, Shaktikanta Nayak, J.P Singh, "Quantum Concepts in Neural Computation", Proceedings of the International Conference on Soft Computing for problem Solving, 2011, Advances in Intelligent and Soft Computing ,Vol. 130, pp. 395-400 , Springer India
- [6] Sitakanta Nayak, Shaktikanta Nayak "A Study on Quantum inspired Hybrid Neural Networks Model", International Journal of Advanced Research in Computer science and Software Engineering, Vol-3, No-06, June-2013
- [7] A. Turing, "On computable numbers with an application to the Entscheidungs-problem," Proc. Lond. Math. Soc. Ser. 2, 42 (1936), 230-65. Also see A. Church, "An unsolvable problem of elementary number theory," American J. of Math., 58 (1936), 345-63.
- [8] F. E. Hohn, Applied Boolean Algebra – An Elementary Introduction, The Macmillan Company, New York, 1966.
- [9] R. P. Feynman, "Quantum mechanical computers," Found. Phys., 16 (1986), 507