GRADIVA

Volume 65 | Issue 07 | July 2026
DOI: 10.5281/zenodo.21701296

Quantum Computing Models for Information Processing: A Comprehensive Study

Sulochana Sonkamble[1] & Balwant Sonkamble[2]

1.Department of Computer Engineering, JSPM Narhe Technical Campus, Pune, (Maharashtra State), India.
2.Department of Computer Engineering, Pune Institute of Computer Technology, Pune, (Maharashtra State), India.
Email: [1]ssonkamble_comp@jspmntc.edu.in, [2]basonkamble@pict.edu
ORCID: [1]0000-0002-6081-8838, [2]0009-0001-8681-6347

Abstract

The quantum computing has been proven to provide the solutions to most complex computations in the world today. Quantum computers can solve the problems with polynomial resources. Quantum information can be encoded in a number of ways, such as spin components of basic particles like photons, electrons, and polarization of photons. The quantum computer uses quantum bits which are superposition of states called as qubits, which is the classical fundamental unit of information. A quantum computer is a regular classical computer with additional ability to control the quantum bits. It is a regular computer connected to a special peripherals and a set of quantum bits simulated to build controller architecture. Two simulators are used for the quantum programming, one is Microsoft Q# and other is IBM Qiskit. Classical language Q # is used to program the quantum controller and its operations such that initializing a qubit, applying a gate on a qubit and measuring a qubit. IBM Qiskit programs simulates the functionality of quantum circuit and creates it's drawing using a graphical interface with qubits, gates, and measurements. Qiskit provides simulators that run quantum circuits using visualization tools that help to understand the effect of our quantum circuits on the quantum states. The teleportation technique allows the quantum information to be communicated at a distance. An unbreakable encryption protocol is built for polarizing photons through the quantum channel. The combination of quantum computing and AI has the potential to drive significant advances across many sectors.

Keywords: *Quantum Computing, Information Processing, Qubit, Cryptography, Q#, Qiskit.*

1. INTRODUCTION

Information theory and quantum mechanics are most influential and revolutionary theories in today's AI powered computer world. It gives a new rise of quantum information theory inspired the novel applications including different algorithms and protocols. Information theory consists of computer science, information sciences and communications with efficient algorithms through robust communication protocols. Quantum computing processes information using quantum physics principles to solve certain problems much faster than classical computers. The quantum computing is based on quantum bits or qubits. Traditional computers the information is represented using bits with a value of either zero or one, whereas a qubit represents values of a zero, a one, or both values simultaneously. Representing thr information in qubits allows the information to be processed in a ways that have no equivalent in classical computing, taking advantage of phenomena such as quantum tunneling and quantum entanglement. As such, quantum computers may theoretically be able to solve certain problems in a few days that would take millions of years on a classical computer [4].

Volume 65 | Issue 07 | July 2026
DOI: 10.5281/zenodo.21701296

In quantum optics, photons, particles of light are used to carry quantum information. Each photon has a polarization — for instance, vertical or horizontal, which can be ascribed with the classic bit states of zero and one, respectively. But polarization can also be in a quantum superposition of these states — essentially zero and one at the same time. Since the means of manipulating the polarization of photons are well understood and easily achievable, optics makes an ideal test-bed for investigating quantum information processing. Some of the first realizations of novel quantum effects, such as teleportation and quantum key distribution, were achieved through optics research.

Quantum computers are programmed using the principles of quantum physics using the property of the polarization of light. Polarized photons of light behave like quantum bits or qubits can be directly used in controlling qubits of quantum computers. A qubit is the fundamental unit of computation in a quantum computer [2]. It plays a role similar to bits in classical computers. A quantum bit has two important behaviors that we can study through photons, Superposition, and Entanglement. Photons are polarized at an angle, when theta is 0, it is called as horizontal polarization and theta is 90, it is called as vertical polarization. In classical computers, the concept of bits is used to represent the boolean values of TRUE and FALSE are encoded as a bit value of 0 and 1, respectively.

Quantum Computing is becoming the next wave of the software industry. Quantum computers are exponentially faster than classical computers of today. A 64-bit quantum computer can process 36 billion bytes of information in each step of computation whereas 8 bytes can process in each step of computation by personal computer. Problems that were considered too difficult for computers to solve, such as simulation of protein folding in biological systems, and cracking RSA encryption, are now possible through quantum computers. Quantum computers are not programmed traditionally but are structured as "circuits" that operate on quantum bits or qubits to perform computations using programming tools Microsoft's Q# and IBM's Qiskit. Quantum computing applications are developed using the popular Shor's algorithm for encryption to protect data on the internet [3]. The content of this paper are broken up into four sections, first section gives introduction, second section emphasize on the material and methods, third section presents the results and discussion and conclusion is presented into final fourth section.

1.1 binary bits and qubits

There exists some kind of well defined logic for every computation. The classical computation is being performed using Boolean logic which uses variables 0 and 1 values.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 is a quantum bit which is represented as a vector in a two dimensional complex Hilbert space. The logic that can be implemented with such qubits is quite distinct from classical Boolean logic and this is what has made Quantum computing exciting by opening new possibilities. In recent years quantum computing becomes a forefront research in the field of algorithms, cryptography and artificial intelligence [1].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

Volume 65 | Issue 07 | July 2026
DOI: 10.5281/zenodo.21701296

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 emphasising the shortcomings of classical gates [20].

1.2 Logical Gates in Classical Computer

In 1936, Scientists, Church and Turing has been introduced a classical computation theory and presented their investigation into the characteristics of computability . Logic gates and logic circuits played a major role in theory of computation to develop the optimal structure of classical logic circuits [20]. 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. Table (1) shows the input - output of the logical gates in classical computer. The notions of boolean functions of classical logic gates: AND, OR, XOR, NAND, NOR are given below in quations (1),, (2), (3), (4) and (5) respectively.

AND-GATE:

It is defined as a Boolean Function:

$$f(x,y) = \begin{cases} 1, if\ x = y = 1 \\ 0, otherwise \end{cases} \text{ (1-a)}$$

The result can be written as:

$$f(x,y) = xy\ (Meaning: product\ of\ x\ and\ y) \text{ (1-b)}$$

OR - GATE:

It is defined as a Boolean Function:

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

The result can be written as:

$$f(x,y) = x + y\ (Meaning: plus\ of\ x\ and\ y) \text{ (2-b)}$$

XOR - GATE:

XOR (Exclusive – OR) is a variant of OR function

It is defined as a Boolean Function:

$$f(x,y) = \begin{cases} 0, if\ x = y = 0\ or\ x = y = 1 \\ 1, otherwise \end{cases} \text{ (3-a)}$$

The result can be written as:

$$f(x,y) = x \oplus y\ (Meaning: plus\ of\ x\ and\ y) \text{ (3-b)}$$

The Exclusive-OR is an interesting gate, when both inputs are 1, the output is zero. In exclusive-OR, interpreted as exclusive-ORing one boolean variable onto another, "exclusive OR B onto A" means the same as A = A XOR B. In other words, A is changed to the result of

Volume 65 | Issue 07 | July 2026
DOI: 10.5281/zenodo.21701296

A XOR B and this is done in a single atomic operation. The value of B is XOR-ed onto the value in A, and the result remains in A.

NAND-GATE:

NAND (NOT-AND) is a universal gate from which any Boolean function can be derived. Let $f(x)$ be a NOT function and $g(x,y)$ be a AND function. The NAND function can be defined as,

$$h = f(g) = \overline{xy} \quad (4)$$

NOR – GATE:

Similarly NOR gate is another universal gate from which any Boolean function can be derived. It can be defined as a function:

$$f(x+y) = \overline{x+y} \quad (5)$$

Table 1 (a): Input-Output of Logical Gates in Classical Computers

Sr. No.	Input		GATE Output				
	x	***y***	**AND**	**OR**	**XOR**	**NAND**	**NOR**
1.	0	0	0	0	0	1	1
2.	0	1	0	1	1	1	0
3.	1	0	0	1	1	1	0
4.	1	1	1	1	0	0	0

1.3 logical gates in quantum computer

The quantum computers perform operations differently based on quantum mechanics laws. Its basic unit is quantum bits, called as qubits. Qubit has a quaternary nature, it exists in the states corresponding to the logical values 0 or 1 like a classical bit, and additional a superposition state. A qubit represents a bit of information that can be both zero and one simultaneously. Thus, a computer working on a qubit rather than a standard bit can make calculations using both values simultaneously. A qubyte, is made up of eight qubits and can have all values from zero to 255 simultaneously. The memory of a classical computer is a string of 0s and 1s, and it can perform calculations on only one set of numbers. The memory of a quantum computer is a quantum state that can be a superposition of different numbers. A quantum computer can do an arbitrary reversible classical computation on all the numbers simultaneously. Specially, performing a computation on many different numbers at the same time and then interfering all the results to get a single answer, makes a quantum computer much powerful than a classical one [19]. The classical signals of voltages are represented by the state 0 and 1 which store one bit information whwreas, two bit states represented by 00,01,10,11 store two bit information, it is 2^n bits. The quantum computers can process n-inputs with only one computational step after the superposition state of n-inputs. Quantum bits are called qubit which used electron states to store the information. The states of the electron are indicated by the spin-up & spin-down and represented as 0 & 1 respectively. The photons are used as a qubit to represent polarization in vertical & horizontal direction. One qubit represents superposition of 0 & 1 states which are denoted by dirac notation, $|0\rangle$ and $|1\rangle$ respectively. 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. Quantum programs contain the equations of a sequence of quantum gates on a circuit. The behavior of these quantum gates is described by matrices of complex numbers [14]. These complex numbers in the matrices affect the probabilities of qubit measurements [20].

Volume 65 | Issue 07 | July 2026
DOI: 10.5281/zenodo.21701296

In quantum computing architectures, an atomic XOR gate is an important component to compute A XOR B is to XOR B onto A in an atomic operation. Exclusive OR gate can be used to perform variable assignment, like A = B. First initialize A to be 0, then Exclusive OR B onto A, this is equivalent to setting A to be A XOR B. At the end of both these steps, the value of B has been assigned to A and this has accomplished to replace assignment with two steps. In first step, initialize to 0 then perform XOR operation. In second step, perform XOR B onto A operation, it can be initialized to 0 very early in a computation before the value of B is available. Finally, the end result is replacement of value of A to value of B. This method of replacing assignment with XOR is very important for quantum computations. Quantum assignment is an irreversible operation and can be performed only once, at the beginning of quantum computations. In XOR single bit operation, entire sequences of bits can be XOR-ed with each other whereas in two bit sequences it results in another bit sequence of the same length. Following example table 1 shows the output bit sequence is obtained by XOR-ing the corresponding bits of the input.

Table 1 (b): XOR Bit Sequences

Input Sequence x	1	0	1	1	0	1	1	1	0	0	0
Input Sequence y	0	1	1	0	1	1	0	1	1	0	1
Output Sequence $x \oplus y$	1	1	0	1	1	0	1	0	1	0	1

2. MATERIALS AND METHODS

2.1 Cryptography with XOR Gate

In real world day-to-day life, it became very essential, to keep private data secure and to protect sensitive information on internet. In view of this, cryptography is used for interaction between sender and receiver, sending secret messages over a insecure channel and the message to be remain a secret. The sender of a secret message can encrypt the message whereas the receiver can decrypt the message [15]. Following table-2 shows, the encription and decryption using XOR gate through a classical communication channel. The XOR operation is used for cryptography to perform two tasks, one is sender encrypts the message called as encryption, which convert the protected message into a sequence of bits. Other task is receiver decrypts the message called as decryption, to get a random sequence of bits that only the sender and the receiver know. This random sequence is a shared secret which is a bit sequence that only know the receiver and sender and operate using the bit sequence on the same message. Finally, the encryption operation is performed on bit sequence using XOR to encrypt the message.

Table 2: Cryptography with XOR

	Encryption										
Input	1	0	0	1	0	1	1	0	1	0	Message encoded as bits
	0	1	0	1	1	0	1	1	0	1	Random sequence of bits
XOR	1	1	0	0	1	1	0	1	1	1	Encrypted message
	Decryption										
Input	1	1	0	0	1	1	0	1	1	1	Received bits, encrypted data
	0	1	0	1	1	0	1	1	0	1	Random sequence of bits
XOR	1	0	0	1	0	1	1	0	1	0	Original message, Decrypted

The encryption is performed by XORing a random bit sequence onto the message. The decryption is performed by XORing the same bit sequence onto the encrypted data. The result of the second XOR is the original message. This is represented by following equation

Volume 65 | Issue 07 | July 2026
DOI: 10.5281/zenodo.21701296

(5), where x is the message encoded as bits, r is the random sequence .

$$x \oplus r \oplus x \oplus r \oplus r = x \quad (5)$$

2.2 Quantum Cryptography

Quantum cryptography allows us to create secure communication protocols that are unbreakable. The cryptography protocol is called BB84, this protocol is based on the quantum properties of light. Light is a stream of photons and a photon behaves like a particle. The light is an electromagnetic wave means the electric and magnetic oscillations are at right-angles along the X-Y plane to the direction of propagation of the wave. Each photon is directed along the z-axis has its own angle, called the angle of polarization, it can be vertical or horizontal. The photon is the system, measured by the polarization angle using a filter which changes the polarization of the photon when it makes it passed the filter. The photons emitted from the light source are having all angles of polarizations as well as vertically polarized and horizontally polarized [18].

The experiment proves that polarizing filters are not really filters but they are quantum mechanical measurement devices and quantum measurements change the state of the systems being measured. When light is polarized at the same angle as the filter, all the lights gets through deterministic behavious. When a photon's polarization is aligned with a polarizing filter, the behavior is deterministic. The photon always passes through. When a photon is anti-aligned with a polarizing filter, the behavior is deterministic. The photon is always blocked for any angle that is not aligned or anti-aligned, the behavior of the photon is probabilistic. Randomness is inherent in the behavior of quantum systems. Encryption with single sequence of bit uses the shared secrets, when any sequence of bits is XORed twice with secret sequence, the original sequence of bits is recovered. If the shared secret is a random sequence of bytes and if each shared secret is used only once, then the produced code is unbreakable. The shared secret should be used only once, if it is used for more than one message, then non-random patterns in the encoded messages will allow the code to be broken. The main objective is to securely establish a shared secret sequence of bytes through a quantum protocol.

2.3 Quantum Teleportation

Sender and receiver are connected by two channels of communication, like internet a classical channel. Both are potentially insecure and anyone can eavesdrop on this channel. There is a quantum channel where sender can send polarized photons to the receiver. Sender first creates a long sequence of random bits and needs to send this sequence to receiver in a secure manner.

Once receiver gets these bits, both can use them as a shared secret in future communication between themselves. This sequence of random bits which sender has created will become the shared secret. The important thing is, not all of the bits which sender has created here will be in the final shared secret. A random fraction of these bits will be lost during transmission when it apply the BB84 protocol. Only a fraction of these bits will reach to receiver and become the shared secret. why some bits are corrupted on protocol.and neither sender nor receiver are able to control it during the transmission [17].

The simple approach to build an unbreakable encryption protocol is use of polarized photons through the quantum channel. Sender encodes the bits with vertical and horizontal polarization, communicates a bit value of 1 with a vertically polarized photon and encodes a

Volume 65 | Issue 07 | July 2026
DOI: 10.5281/zenodo.21701296

bit value of 0 as a horizontally polarized photon. Using this encoding protocol, sender can send the sequence of bits to receiver. At this stage, anyone eavesdropping on the quantum channel can read the data being sent without either sender or receiver realizing that their security has been compromised. If the photon passes through the filter, then sender had sent a value 1. If the photon is blocked, then sender had sent a value 0. Next a new photon is created that encodes the same bit that sender had sent to receiver. Receiver receives the photon sent through filter, cannot be realized that it has intercepted and resent the entire sequence of bits message that sender sent. It indicates that this is not a secure protocol means all data can read then resend it without being detected. Therefore , it is essential to establish a secure protocol at receiver. The security of the protocol is based on quantum behavior of photons of laws of physics [11].

Some of the bits will be corrupted and lost in transmission. These bits for which receiver has guessed wrong are the ones that get corrupted and are lost in transmission. To prevent the loss, sender modifies a protocol, sender chooses to encode a randomly chosen 50% of the bits in a sequence with vertical polarization for bit value 1and horizontal polarization for bit value 0. For the other 50% , sendor encodes bit values of 0 with photon set at polarized 45^0 to the left and encodes bit values of 1 with photon set at polarized 45^0 to the right. The choice of which encoding to use for each bit is made at random. There is a 50% probability that sender will choose the vertical & horizontal encoding and there is a 50% probability that sender will choose the 45% left and right encoding.

Table 3: Encoding Data in Photon Polarization

0	1	0	0	1	1	1	0	1	0	1	1	Sender's sequence
+	+	X	+	X	X	+	X	+	X	X	X	Randomly chosen encoding scheme
-	|	\	-	/	/	|	\	|	\	/	/	Polarization angles of photons sent by sender
Quantum Channel												
X	+	+	X	X	X	+	X	X	+	X	+	Receives guess of encoding scheme
?	1	?	?	1	1	1	0	?	?	1	?	Receiver decodes photons based on his guess of encoding scheme.
	√			√	√	√	√			√		Receivers correct guess

Table (3) shows the encoding data in photon polarization angles, top row shows a bit of sequence to be send by sender, below that a random choice of encoding is given. There is one choise for each of the bit, + symbols represent the verticle and horizontal encoding scheme. When a value 0 is encoded with vertical and horizontal encoding scheme, the photon is horizontally polarized and represented by - symbol. When a value 1 is encoded with vertical and horizontal encoding scheme, the photon is vertically polarized and represented by | symbol. Other half of bits are encoded using 45^0 left and right polarization which is represented by symbol X, symbol \ indicates bit value 0 in 45^0 encoding and the photon is polarized 45^0 to the left. Symbol / indicates a bit value 1 in 45^0 encoding and the photon is polarized 45% to the right.

On the receiver side, yet to know the encoding data that is used for each bit and therefore receiver makes random guesses, 50% guesses are correct and remaining 50% guesses are wrong. Receiver decodes photons based on his guess into vertical-horizintal encoding scheme, and for these bits, it guesses 45^0 left - right encoding. In table (3), symbol √ indicates that receiver decode the photons correctly and aligns the polarizing filter according to the guess and the photon behaves in a deterministic manner which enables the receiver to decode correctly [4]. Remaining guesses are wrong and it will align the polarizing filter at a wrong angle and the behaviour of the photon is random. Randomly some photons get through

Volume 65 | Issue 07 | July 2026
DOI: 10.5281/zenodo.21701296

the filter and some photons will not pass the filter [16].

Therefore receiver attempts to decode its bits using wrong guesses results in a random set of bits and these bits are corrupted and lost in transmission. Sender reveals the correct encoding after receiver receives the photons. If data is corrupted, sender and receiver discard the entire sequence and try again with a new set of data, eventually both are having the shared sequence code. BB84 protocol allows the sender and receiver to share a secrete key over insecure communication channel, since they have a shared secret, both can use 1-use-shared-secret encryption which is unbreakable. The security of the protocol is based on quantum behavior of photons and it is based on the laws of physics which remains secure in future. Quantum computers can crack RSA encryption using quantum properties.

2.4 Mathematical Representation of Quantum States

The quantum system is consists of Qubit, which is equivalent to the bit used in classical computers. The qubit has two states ON and OFF or a superposition of ON and OFF. Quantum system in superposition requires more than one numbers to define it. The quantum system can be represented mathematically using a matrix algebra known as a linear algebra and quantum states are represented by complex numbers of column matrices with dimension 4X1. The changes to this state are represented by multiplying the column matrix by a transformation matrix [5]. A quantum state without superposition is represented by a single number in the column matrix, all others are 0. Square magnitudes of the numbers in the column matrix represent probabilities.

Following figure (1) shows the matrices for quantum states, the transformation matrix of complex numbers and quantum states without superposition.

$$\begin{bmatrix} s+ti \\ u+vi \\ w+xi \\ y+zi \end{bmatrix} \qquad \begin{bmatrix} a & b & c & d \\ e & f & g & h \\ i & j & k & l \\ m & n & o & p \end{bmatrix} \qquad \begin{bmatrix} s+ti \\ 0 \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} 0 \\ u+vi \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ w+xi \\ 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ y+zi \end{bmatrix}$$

Fig 1: (i) Quantum states (ii) Transformation matrix of complex numbers (iii) Quantum states without superposition

When electrons move up and down in a magnetic field, this property of electrons is called spin. The spin can be used as the quantum equivalent of a bit in classical computers. When the spin is up, it is equivalent of a bit 1 and when spin is down, it is equivalent of a bit 0 . The quantum equivalent of a bit is called a qubit and this is a quantum bit. The probability of measuring spin up OR down is 1 and probability of measuring spin up AND down is 0. This property can be used as the basis for defining the states of the spin qubit represented by matrix given below in figure (2). If a qubit is in the up state, corresponding to a classical bit being 1, the matrix representation of the state will be [0 1]. If a qubit is in the down state, corresponding to a classical bit being 0, the matrix we use will be [1 0].

$$\begin{bmatrix} 0 \\ 1 \end{bmatrix} \qquad \begin{bmatrix} 1 \\ 0 \end{bmatrix} \qquad \begin{bmatrix} 1 \\ 0 \end{bmatrix} \qquad \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

Fig 2: Spin Qubit States: (i) UP state. (ii) DOWN state. Reversible state transformation:(iii) Qubit in state |0> (iv)Qubit in state |1>

Volume 65 | Issue 07 | July 2026
DOI: 10.5281/zenodo.21701296

In classical physics, the location-state of a particle can be specified in terms of the three axes X, Y and Z, the three coordinates are usually written as three numbers enclosed (x,y,z). Whereas in quantum physics, the numbers which describe a state are written as a column matrix, it is called a vector. A vector represents a qubit, has two elements, the upper number is the square-root of the probability of being measured in the Qubit-Off or Spin-Down state. The lower number is the square-root of the probability of being measured in the Qubit-On or Spin-Up state.

2.5 Superposition And Entanglement In Quantum Computing

When the polarization of photon is not aligned and not anti-aligned with a measurement apparatus then photon polarization is in superposition, and the measurement on a superposed state has probabilistic behaviour. When photon polarization is aligned then qubit value is 1 value [10]. When photon polarization is anti-aligned then qubit value is 0 whereas when photon polarization is between 0 and 90 degrees, the qubit is in superposition and the result of measurement is probabilistic. In Reversible State Transformations, electron spin were used as the qubit in a quantum computer Up = 1 and Down = 0, $[1\ 0]^T$ represents the qubit is in state 0, and $[0\ 1]^T$ represents the qubit is in state 1. When, the qubits are sent as input to a NOT gate, it transforms into a reversible form. It changes bit value 1 to bit value 0 and bit value 0 to bit value 1. Applying a transformation is just matrix multiplication, that is, $[1\ 0]^T$ has been transformed to $[0\ 1]^T$ means the qubit value of 0 has become qubit value of 1. Similarly, apply the NOT operation to a qubit value 1, become the qubit value 0. In case of two qubit systems, the possible combinations are 00, 01, 10, and 11.

The state vector will have four elements, one for each combination given further steps i, ii, iii and iv, represented in figure (3). Step (i) when both qubits are 0, the ket vector is written as |00>, step (ii) , when first qubit is 0, and the second qubit is 1, ket vector is written as |01>, step (iii) when the first qubit is 1 and the second is 0, ket vector is written as vector |10>, and step (iv) when both qubits are 1, the ket vector is written as |11>. The measurements and reversible transformations are performed on multi-qubit states in the same way of single qubit states. The superposition state can be written in ket vectors as 1 by root 2 times (|00> plus |11>). If the two qubits are 0, then the probability of the measurement is the square magnitude of number which is 0.5. Similarly if first qubit is 0, and the second qubit is 1, then probability is 0. If first qubit 1, second is 0, then probability is 0. If both qubits are 1, then the probability is square magnitude of 1 by root 2 which is 0.5.

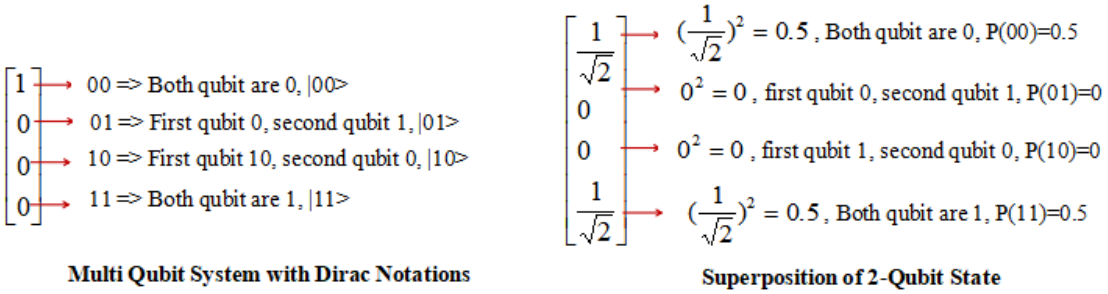

Fig 3: Multi-Qubit System and Superposition States

In ket state vector of a two qubit system, the probability of both qubits being 0 is 50% whereas both qubit being 1 is 50%. Whenever, two qubit have the same value then the value of one qubit is connected to the value of the other qubit and these two qubits are entangled. Qubit entanglement behaves the same as photon entanglement. If two photons are entangled,

Volume 65 | Issue 07 | July 2026
DOI: 10.5281/zenodo.21701296

then after measurement, they are independent of each other and the effect of entanglement is seen in the probability values. If the photon polarization is aligned, then qubit value is 1 whereas if the photon polarization is anti-aligned, then the qubit value is 0. Then qubit is entangled and qubit entanglement behaves same as photon entanglement.

3. RESULTS AND DISCUSSION

3.1 Quantum Computing Model

In quantum computation, qubit state is initialized to 0, it is irreversible, the final output is obtained from the quantum computation which is irreversible and the steps in between are reversible. Figure 4 shows the quantum computations for 4-qubits input and 2-qubits output. Initial values of the bits are set on the left side indicates, the initialization is an irreversible operation and it is performed before quantum computations begin. During the quantum computations, values cannot be assigned to any of the bits as a assignment which is an irreversible operation, so it cannot be used in quantum computations [5]. XOR gate is used onto a qubit to be initialized to 0 value. Whatever a value used with exclusive OR with a bit , it has been initialized to a value zero. Quantum computations are performed with the pattern of four bits of input and produces two bits of output as shown in figure (4). The four bits of input are initialized to whatever the input. The values cannot be assigned to the output because that would be irreversible, instead XORed values onto the output qubit. XOR is a reversible operation and can perform it during quantum computations [20]. If the output qubit are 0 at the starting, then the XOR operation is equivalent to an assignment. Both the input qubit and the output qubit are initialized in the initialization step. Two qubit are allocated at the initialization tstep to store the output and remaining 4-qubit are initialized to the input value. During the quantum computation operation, it will be XORed the output onto the bits and XOR is a reversible operation. When computation has completed, these 2-qubit are measured to produce the output.

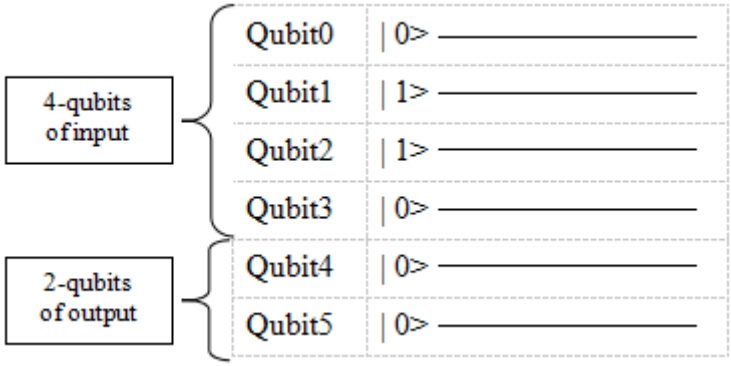

Fig 4: Quantum Computations for 4-Qubit Input and 2-Qubit Output

In quantum computations, fanout gate is used to copy the output to multiple subsequent inputs and copy qubit values 1 or 0 exactly. Most quantum computers have very few qubits as its very expensive piece of hardware, therefore it is prefered to reuse qubits during computations [6]. In quantum computation circuit model, qubit cannot be reinitialized, as it is an irreversible operation which can be done at the beginning of computation. Qubit can be reused by reversing the computations that have been performed on it to return its original state. After a qubit state has been reversed, it can reuse it for further computations. Figure (5) shows the reversible gates with ancilla bit and actions performed by these gates on the ancilla

Volume 65 | Issue 07 | July 2026
DOI: 10.5281/zenodo.21701296

bits are reversed after the target bit has been set. The action of reversing the ancilla bits back to their initialized state allow them to be used again later in the computation.

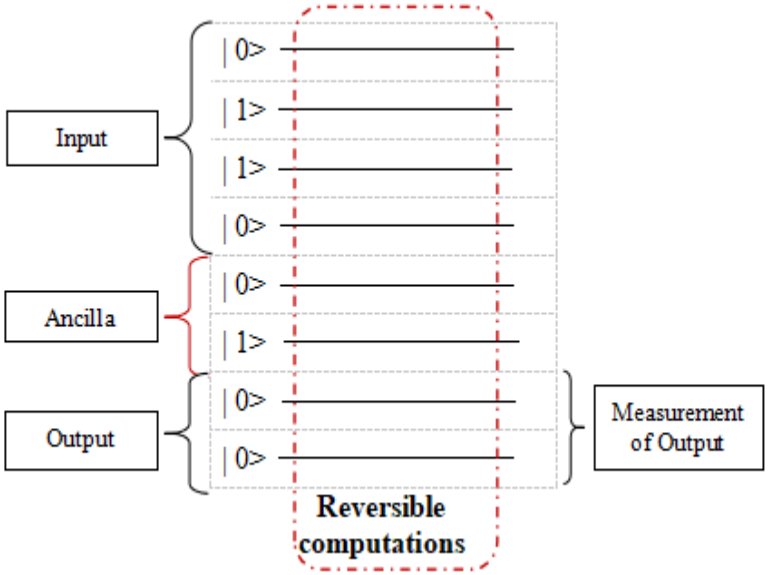

Fig 5: Reversible Gates with Ancilla - Auxillary Bit

In the classical computing there is only one reversible operation performed on bits in NOT gate whereas in the quantum computing, more operations are possible. Following equations (6-a) and (6-b) shows a NOT gate is represented on the computations and the reversible quantum operations are mathematically modeled as a matrix multiplications [12]. Following matrix represents NOT gate appled on a qubit value of 0 that is ket |0> or $[1\ 0]^T$ vector, where X is the NOT gate.

$$X|0\rangle = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}\begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} = |1\rangle \text{ (6- a)}$$

$$X|0\rangle = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}\begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} = |0\rangle \text{ (6-b)}$$

When multipled these matrices , it produces the result $[0\ 1]^T$, that is ket |1> or qubit value 1, NOT matrix has converted |0> to |1>. In next step, NOT operation is applied on qubit value 1, that is ket |1> or $[0\ 1]^T$. After multiplication of these matrices, the produced result is $[1\ 0]^T$ which is qubit 0. Following equation (7) shows an unknown |x> is restored after two NOT operations and two NOT gates used in sequence will be cancelled each other. NOT operation applied twice is the NOT matrix multiplied by itself, and the result of this multiplication is the identity matrix, that is applying the NOT operation twice is the same as doing nothing.

$$XX = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = I \text{ (Identity) (7)}$$

Quantum computers can transform one superposed state to another superposed state and they can transform |0> or |1> to superposed state. In classical logic circuits only NOT gate operate on a single bit whereas in quantum computing there are more single qubit operations

Volume 65 | Issue 07 | July 2026
DOI: 10.5281/zenodo.21701296

are possible just changing ON / OFF because qubits can be in superposition [7]. All the gates have effects on qubits in superposition states. Three NOT gates are denoted in following quation (8-a). Pauli operations on NOT gate invert themselves are represented in equation (8-b) and (8-c) as a resulting identity matrix when operations performed twice on it.

$$X=\begin{bmatrix}0 & 1\\ 1 & 0\end{bmatrix}\ Y=\begin{bmatrix}0 & -i\\ i & 0\end{bmatrix}\ Z=\begin{bmatrix}1 & 0\\ 0 & -1\end{bmatrix} \text{ (8- a)}$$

$$YY=\begin{bmatrix}0 & -i\\ i & 0\end{bmatrix}\begin{bmatrix}0 & -i\\ i & 0\end{bmatrix}=\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}=I \text{ (Identity) (8-b)}$$

$$ZZ=\begin{bmatrix}1 & 0\\ 0 & -1\end{bmatrix}\begin{bmatrix}1 & 0\\ 0 & -1\end{bmatrix}=\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}=I \text{ (Identity) (8-c)}$$

The operations S and T are called as square root of NOT, if this operation applied twice then it works same as NOT operation and results in NOT matrix, equations (9-a) and (9-b) shows the square root of NOT operation.

$$\sqrt{X}=\sqrt{NOT}=\frac{1}{2}\begin{bmatrix}1+i & 1-i\\ 1-i & 1+i\end{bmatrix} \text{ (9-a)}$$

$$\frac{1}{2}\begin{bmatrix}1+i & 1-i\\ 1-i & 1+i\end{bmatrix}\times\frac{1}{2}\begin{bmatrix}1+i & 1-i\\ 1-i & 1+i\end{bmatrix}=\begin{bmatrix}0 & 1\\ 1 & 0\end{bmatrix} \text{ (9-b)}$$

The Hadamard gate is considered to be an important gate as it accepts a state |0> or |1> and changes it to a superposition state H |0> and H |1> as shown in following equations (10-a), (10-b) and (10-c).

$$\frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\ 1 & 1\end{bmatrix} \text{ (10-a)}$$

$$H|0>=\frac{1}{\sqrt{2}}(|0>+|1>) \text{ (10-b)}$$

$$H|1>=\frac{1}{\sqrt{2}}(|0>-|1>) \text{ (10-c)}$$

Two qubits system is represented by a vector that has 4 elements and a gate that operates on 2 qubits modeled by a 4 by 4 matrix. The controlled NOT gate is a two qubit gate with two input and one output as shown on a quantum circuit in figure (6-a) and a CNOT matrix and 2-qubit state vectors in figure (6-b). The behavior of a controlled NOT gate is the x control means when x is 1, then the output is NOT of y and When x is 0, then the output is y. The value of x controlled when NOT operation is applied on y. As shown in figure, when x is 1, NOT operation is applied on y and output is x XOR y whereas when x is 0 output is y. When CNOT matrix is multiplied by each of these two qubit state vectors, four transformed results are produced; |00> transformed to |00>, |01> transformed to |01>, |10> transformed to |11>, |11> transformed to |10>. The controlled NOT gate is the inverse of itself, when CNOT matrix is multiplied by itself, produced result is a 4X4 identity matrix.

Volume 65 | Issue 07 | July 2026
DOI: 10.5281/zenodo.21701296

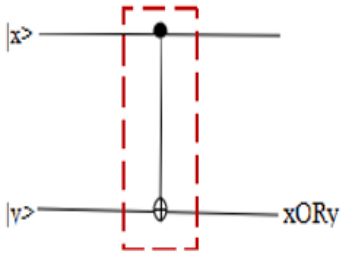

Fig 6 (a): Controlled CNOT Gate

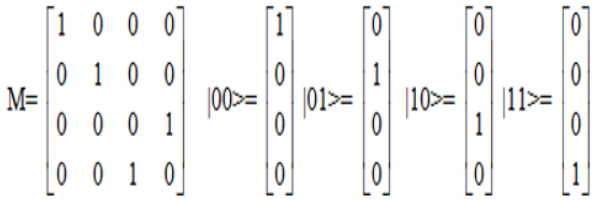

Fig 6 (b): CNOT Matrix and 2-Qubit State Vectors

3.2 Taffoli Gate as a Quantum Equivalent

Toffoli gate consists of 3 qubit gates, among these, 1 control bit is inverted and 2 bits are placed one after another controlled NOT gates. Following figure (7-a) shows the Taffoli Gate with CCNOT for two input and one output, figure (7-b) shows three input and one output. It is represented in following equation (11-a) is the first output and equation (11-b) is final output.

$$zXORxANDy \quad \text{(11-a)}$$

$$zXORxANDyXORxANDy \quad \text{(11-b)}$$

Toffoli Gate CCNOT is a 3 qubit system, which contains 2^3 elements in the state vector and it's model is 8X 8 matrix, two input one output taffoli gate shown in figure (7). Toffoli gate with 3 inputs and one output Toffoli gate can be constructed with 2 input gates as a building block.

The first Toffoli gate places an intermediate value in this qubit, this value is used in the second Taffoli gate to obtain the final result in this qubit. As qubits are precious, it should not be wasted anywhere during the computation. Value |0> has been changed to an unknown state after the first Toffoli gate and third Toffoli gate has been added to uncompute the qubit and reverse it to the initial state.

The Toffoli gate is its own inverse, therefore applying the gate again will reverse the change made by the first Toffoli gate. After the third Toffoli gate,the value of this qubit has been uncomputed back to |0> which can used this |0> in further calculations. Qubits which are set to constant values and used in intermediate calculations are called as ancilla.

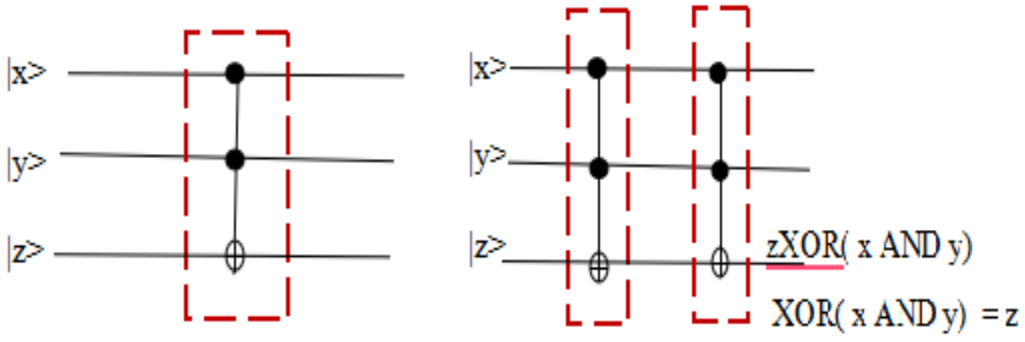

Fig 7 (a): Taffoli Gate CCNOT – Two input& one output, Three input & one output

Volume 65 | Issue 07 | July 2026
DOI: 10.5281/zenodo.21701296

$$\begin{array}{c} \\ 000 \\ 001 \\ 010 \\ 011 \\ 100 \\ 101 \\ 110 \\ 111 \end{array} \begin{array}{c} \begin{array}{cccccccc} 000 & 001 & 101 & 011 & 100 & 101 & 110 & 111 \end{array} \\ \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \end{array} \quad |000\rangle = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad |001\rangle = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad \ldots\ldots \quad |111\rangle = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

Fig 7 (b): Taffoli Gate CCNOT – (a)Taffoli Matrix Model (b) Taffoli State Vectors

Any logical expression can be represented with NAND gate in digital circuits; therefore it is called as a universal gate [8]. Toffoli gate is also a universal gate which can be configured to behaves as a NOT gate, when both control qubits are set to 1 whereas it can behaves as an AND gate if third qubit is set to |0> and first two qubit inputs are set to 1 as shown in figure (6-a) and (6-b) respectively.

Toffoli gate is configured to produce fanout, |y> input is set as two separate output qubit and first qubit is set to |1 shown in figure (6-c), Such taffoli gate is called as quantum equivalent of a universal gate.

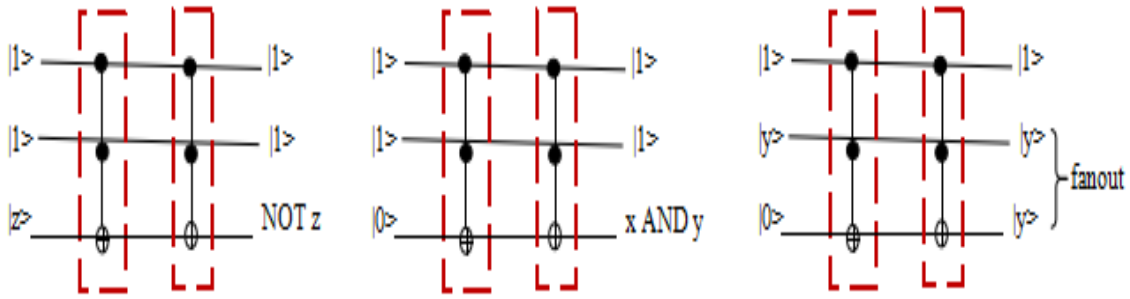

Fig 7 (c): Taffoli Gate Quantum Equivalent of Universal Gate - NOT Gate , AND Gate , Fanout

3.3 Fredkin Gate

Fredkin gate is a universal quantum gate, configured in such a way that qubit |x> is the controller and two qubits |y> and |z> are controlled by x, shown in figure (7-d). When x is set to 0 then two outputs y and z remain the same whereas x is 1, then y and z are swapped to exchange their values.

The Fredkin gate matrix is represented in figure (7-d). When Fredkin gate configured in such a way that, y is set to 1 and z is set to 0 then fanout configuration set to x is 0 or 1, then the value of x will be replicated in two of the outputs, x and z and NOT x will be available in output y, shown in figure (7-d).

Volume 65 | Issue 07 | July 2026
DOI: 10.5281/zenodo.21701296

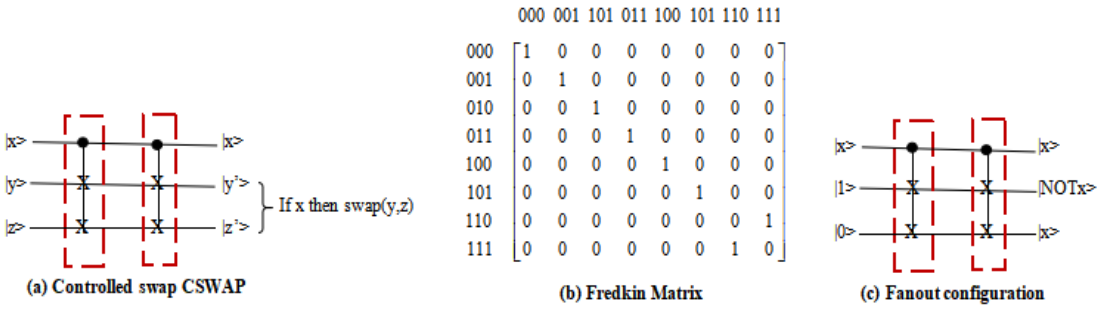

Fig 7 (d): Fredkin in Fan-out configuration:

When Fredkin gate is configured in such a way that x value is set to a superposition state by equation (12), and the output is a Fredkin matrix state vector shown in figure (8) as a result of Fanout on superposition state vector. It gives the output vector by equation (13), calculated by multiplying Fredkin matrix with state vector, which is an entangled state. The quantum gates are designed to be intuitive when qubit are set to 0 or 1, and when the inputs are superposed or entangled, the output can be weird.

$$x = \frac{1}{\sqrt{2}}(|0> + |1>) \tag{12}$$

$$\frac{1}{\sqrt{2}}(|010> + |101>) \tag{13}$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} 0 \\ 0 \\ \frac{1}{\sqrt{2}} \\ 0 \\ 0 \\ \frac{1}{\sqrt{2}} \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \frac{1}{\sqrt{2}} \\ 0 \\ 0 \\ \frac{1}{\sqrt{2}} \\ 0 \\ 0 \end{bmatrix} = \frac{1}{\sqrt{2}}(|010> + |101>)$$

Fig 8: Fanout on superposition state vector

3.4 Discussion of Quantum Programming Tools

A quantum computer is a regular classical computer with additional ability to control the quantum bits. It is a regular computer connected to a special peripherals and a set of quantum bits simulated to build a controller architecture. In this paper two simulators presented, one is Microsoft Q# and other is IBM Qiskit.

3.4.1 Q# Control System:

The controller manages the qubits of the quantum computer and it behaves like a classical computer capable to perform all operations which can be performed by classical computers. Q# controller consists of a memory, input device, output device and a controller processing unit. It is differentiated in addition to classical operations, the controller has a

Volume 65 | Issue 07 | July 2026
DOI: 10.5281/zenodo.21701296

library of operations that control the quantum bits. A classical language Q# is used to program this controller and an additional operation that initialize a qubit, applying a gate on a qubit and measure a qubit [21].

The software of the quantum computers contains the variables, for loops, function calls and other features found in the languages of a classical computers, additionally it contains a library of operations to control the quantun bits. The Q# execution model of a quantum circuit consists of two qubits, Hadmard gate to apply on first qubit to place it in superposition and next control NOT gate to apply on both qubits. This model is implemented in Q# is different from the circuit diagram model. In circuit diagram implicit time-sequence of operations are applied on qubits in a sequence, from left to right. Whereas in the Q# code is explicitly sequential and operations are applied on qubits using code.

A single operation is performed on the qubits because the controller behaves like a single threaded CPU. Parallel quantum operations could be performed in parallel to speedup the performance. Quantum computers has no any significant speedup from the parallelism therefore it follows the sequential model of execution provided by Q#. The speed up of quantum computing comes from the ability of qubits to be in superposed and entangled states.

3.4.2 Overview of 4-Qubit Simulation Framework

The Q# code allocates four qubit and runs a computation on it ten thousand times. After each computation, it measures the qubit and determines if the qubit are in configuration such as 0000 or 0001 or 0010, and so on. The advantage of qubit is that they can be superposed and entangled. When qubit are entangled and superposed, their behavior is a complex [9]. Such complex behavior of superposed and entangled qubits are the foundation of the power of quantum computing.

3.4.3 Significance of Superposition and Entanglement in Q#

The effects of superposition on qubit gates are initialized by assigning qubit to 0. The Hadamard operation is applied on qubit 0, it converts the qubit 0 value into a superposed state of $[1/\sqrt{2} \quad 1/\sqrt{2}]$. The superposed state as the control in a CNOT operation is applied on qubit 0 and 1. The output indicates that half the time qubits 0 and 1 indicates both 0. The other half of the time, qubit 0 and 1 indicates both 1. The configurations where qubits 0 and 1 are not equal to each other in the output.

The state vector of the four qubit system shows the square magnitudes, then only the 0000 and 0011 configurations occurs with probability of half [13]. This state vector represents an entangled state.The qubits 0 and 1 are entangled so that they are always equal to each other where qubits 0 and 1 are not equal to each other. This is called the bell state for the qubit 0 and 1 to be considered for information alert schedule learning time.

3.4.4 Overview of IBM Quantum Computer

IBM developed the quantum computers which are capable to design a quantum circuit by dragging and dropping from the gates and operations. The limited physical connectivity of the qubit in IBM computer is possible, and not for every combinations of qubit gates. It is possible to initialize a quantum program and simulate to test the results in a graphical charts. The qubit configuration 00000 and 00011 are equally probable for the qubits 0 and 1 are set in the bell state where they are measured to be equal to each other.

Volume 65 | Issue 07 | July 2026
DOI: 10.5281/zenodo.21701296

3.4.5 Quantum Programming and Algorithms With IBM Qiskit

Qiskit is Quantum Information Software Kit by IBM, contains a python library for running quantum programs. Qiskit programs simulates the functionality of quantum circuit and creates it's equivalent drawing using a graphical interface with qubits, gates, and measurements. Qiskit provides simulators that runs quantum circuits using visualization tools that help to understand the effect of our quantum circuits on the quantum state [22]. These visualization tools are playing a role of a quantum debugger. Qiskit's built-in implementation tools are useful to run complex algorithms, such as Shor's algorithm. The quantum computers have limitations, that the quantum bits can not be entangled with each other, whereas quantum algorithms design with all the qubit can be entangled with each other. Qiskit provides a transpiler which can convert idealized quantum programs which run on the real-world quantum computers.

The circuits are created new kinds of custom gates, shown in figure (9) along with the implementation and drawings. A quantum circuit with three qubit is created with parameterized circuit template using Qiskit tool with the number of parameters replaced by actual values [23]. Figure (9) shows, a line initialized the state vector of the three qubit with eight number values in the state vector, initializing all three qubits from a state 1 and last number represents the qubit state 111. Three parameters, angle0, angle1, and angle2 are represented in a template circuit and place the Hadamard gate on the three qubits. The phase-change gates are placed on the three qubits to change the angle of the qubit which is used to draw the template parameter circuit.

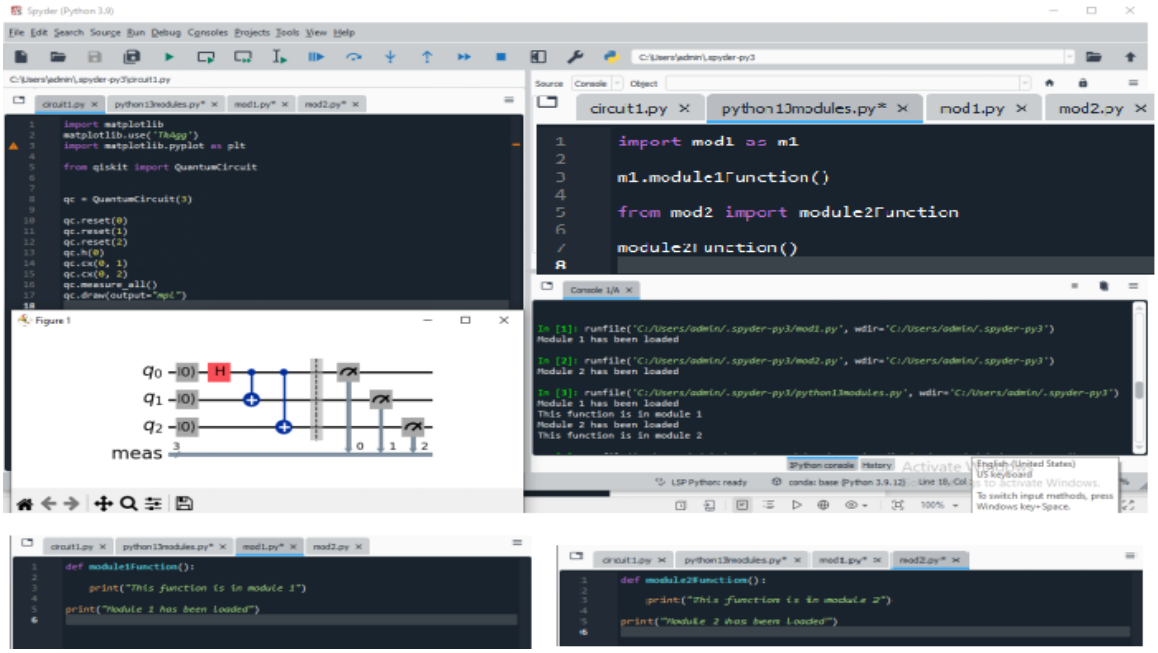

Figure 9: Parameterized template circuit in Qiskit

4. CONCLUSION

Quantum computing works on the principles of quantum mechanics, including superposition, entanglement, and quantum interference. It process the information using qubit, which can exist in multiple states 0, 1 and both simultaneously. This fundamental property enables quantum computers to address complex computational problems more

Volume 65 | Issue 07 | July 2026
DOI: 10.5281/zenodo.21701296

efficiently, with polynomial resource requirements, making quantum computing a promising solution for challenges that are intractable for classical computers.This paper presented the core concepts of quantum computing and quantum information processing, focusing on the representation, manipulation, and measurement of quantum information based on the gates computations. Various methods of encoding quantum information, such as spin states and photon polarization, are presented to highlight the flexibility and physical realizations of the qubit. The role of quantum gates, including the Hadamard gate, in creating superposition and facilitating quantum operations is emphasized, demonstrating how quantum logic forms the basis of quantum algorithms.

This paper focus on the significance of quantum simulation platforms such as Microsoft Q# and IBM Qiskit. These tools provide powerful environments for designing, simulating, and visualizing quantum circuits, enabling researchers to experiment with quantum algorithms without requiring access to physical quantum hardware. The visualization features offered by Qiskit enhance conceptual understanding by illustrating quantum state evolution and measurement outcomes, to support the future study and research in quantum computing. The paper presents the practical applications of quantum information processing, particularly quantum teleportation and quantum cryptography. Quantum teleportation enables the secure transmission of quantum states across distances using entanglement, while polarization-based quantum cryptographic techniques offer theoretically unbreakable encryption for secure communication. Finally, the convergence of quantum computing and artificial intelligence was identified as a transformative research direction with the potential to accelerate information processing, optimization, and intelligent decision-making. As quantum technologies mature, a solid foundation in quantum computing principles will be essential for developing future secure, scalable, and intelligent computational systems.

Future research will target the development of quantum systems that can solve large-scale, practical problems. Areas of impact will be drug discovery, financial modelling, energy optimization, and climate modelling. By simulating molecules and optimizing complex systems, quantum computing can make great contributions toward personalized medicine, sustainable energy technologies, and predictive analytics. Continued success in quantum computing will hinge on borderless and multidisciplinary collaboration. Therefore, governments, academia, and industry must collectively over-come technical challenges, ensure the establishment of inter-national standards, and make sure that integration into existing systems is practical and efficient. International partnerships will accelerate development and help ensure that these benefits are realized on a global scale. In summary, although the concept of quantum computing is very young, it could easily solve some of the world's most challenging problems. Future research will target the development of quantum systems that can solve large-scale, practical problems. Areas of impact will be drug discovery, financial modelling, energy optimization, and climate modelling. By simulating molecules and optimizing complex systems, quantum computing can make great contributions toward personalized medicine, sustainable energy technologies, and predictive analytics

The future research in quantum computing will be continued for multidisciplinary collaborations in the wide areas like government, academia and industry by integrating into existing systems to accelerate developments and ensure benefits on global scale. The major areas of development will be to solve large scale practical problems and its impact will

Volume 65 | Issue 07 | July 2026
DOI: 10.5281/zenodo.21701296

continue to reshape the face of financial modelling, climate modelling, predictive analytics, and sustainable energy technologies.

5. DECLARATIONS

5.1 Use of AI Technology

The authors not used AI-based tools for editing purposes.

5.2 Conflicts of Interest Declaration

All authors declare that they have no conflicts of interest.

5.3 Informed Consent Declaration

No, my research did not involve human participants.

5.4 Funding

This work received no external funding.

5.5 Data Availability Statement

No new data were generated or analyzed in this paper.

5.6 Ethics and Consent Human Participants

No human participants were involved in study, Not Applicable.

5.7 Consent to publish Identifiable Images

No identifiable images are present.

References

1) Sukhpal Singh Gill, Rajkumar Buyya: Transforming research with quantum computing. Journal of Economy and Technology, 4, 1-8, 2949-9488, (2026).doi:10.1016/j.ject.2024.07.001.

2) Lund, B.D.; Shahriar S.: Quantum Computing: A Concise Introduction. Encyclopedia, 5, 173. (2025). doi:10.3390/encyclopedia5040173.

3) David Barral, F. Javier Cardama, Guillermo Díaz-Camacho, Daniel Faílde, Iago F. Llovo, Mariamo Mussa-Juane, Jorge Vázquez-Pérez, Juan Villasuso, César Piñeiro, Natalia Costas, Juan C. Pichel, Tomás F. Pena, Andrés Gómez :Review of Distributed Quantum Computing: From single QPU to High Performance Quantum Computing. Computer Science Review, 57, 100747, 1574-0137, (2025). doi:10.1016/j.cosrev.2025.100747.

4) Giovanni Acampora, Andris Ambainis, Natalia Ares, et al. : Quantum computing and artificial intelligence: status and perspectives. arXiv:2505.23860, 1-33. (2025). doi:10.48550/arXiv.2505.23860.

5) Christopher Columbus Chinnappan, Palani Thanaraj Krishnan, Elakiya Elamaran, Rajakumar Arul, T. Sunil Kumar. Quantum Computing: Foundations, Architecture and Applications. Engineering Reports, John Wiley & Sons Ltd., 7(8). (2025). doi:10.1002/eng2.70337.

Volume 65 | Issue 07 | July 2026
DOI: 10.5281/zenodo.21701296

6) Pan, X., Cao, X., Wang, W. et al.: Experimental quantum end-to-end learning on a superconducting processo. npj quantum information, 9(18). (2023). doi: 10.1038/s41534-023-00685-w

7) Cerezo, M., Verdon, G., Huang, HY. et al.: Challenges and opportunities in quantum machine learning. Nature Computational Science, 2, 567–576. (2022). doi:10.1038/s43588-022-00311-3.

8) Jose D. Martín-Guerrero, Lucas Lamata: Quantum Machine Learning A tutorial. Neurocomputing, 470, 457-461, 0925-2312. (2022). doi:10.1016/j.neucom.2021.02.102.

9) Ciaran Hughes • Joshua Isaacson, Anastasia Perry , Ranbel F. Sun, Jessica Turner: Quantum Computing for the Quantum Curious, Springer, 1, 978-3-030-61601-4, (2021). doi:10.1007/978-3-030-61601-4.

10) Huang, HY., Broughton, M., Mohseni, M. et al. : Power of data in quantum machine learning. Nature Communications, 12, 2631. (2021). doi:10.1038/s41467-021-22539-9.

11) Ciaran Hughes, Joshua Isaacson, Anastasia Perry, Ranbel F. Sun, Jessica Turner :Quantum Computing for the Quantum Curious. Springer, 978-3-030-61601-4. (2021).

12) Jacob Biamonte, Peter Wittek, Nicola Pancotti, Patrick Rebentrost, Nathan Wiebe, Seth Lloyd: Quantum Machine Learning. (2018). arXiv:1611.09347v2.

13) Sergio Boixo, Troels F. Ronnow, Sergei V. Isakov, Zhihui Wang, David Weker, Daniel A. Lidar, John M. Martinis, Matthias Troyer : Quantum Annealing with more than one Hundred Quibits. (2013).

14) Shaktikanta Nayak, Sitakanta Nayak, J. P. Singh : An Introduction to Basic Logic Gates for Quantum Computer. International Journal of Advance Research in Computer Science and Software Engineering, 1(10), 163-171, 2277128. (2013).

15) Simon J. Devitt, William J. Munro, Kae Nemoto : High Performance Quantum Computing. special issue on Quantum Information Technology, Progress in Informatics, 8, 49-55. (2011). doi:10.2201/Niipi.2011.8.6.

16) Mingsheng Ying : Quantum Computation, quantum theory and AI. Artificial Intelligence, Elsevier, 174, 162-176. (2010). doi 10.1016/j.artint.2009.11.009.

17) Y. Kanamori, S. M Yoo, W. D. Pan, and F. T. Sheldon : A short survey on quantum computers. International journal of computers and applications, 28(3), 227-233. (2006).

18) Mark Oskin, Francois Impens, Tzvetan Metodiev , Andrew Cross, Frederic T. Chong, Issac L. Chaung, John Kubiatowicz : Towards a Scalable , Silicon – Based Quantum Computing Architecture. IEEE Journal of Selected Topica in Quantum Electronics, 9(6). (2003).

19) Micheal A. Nielsen, Issac L. Chuang : Quantum Computation and Quantum Information. Cambride University Press. (2003).

Volume 65 | Issue 07 | July 2026
DOI: 10.5281/zenodo.21701296

20) Ivan B. Djordjevic : Quantum machine learning, Quantum Communication, Quantum Networks, and Quantum Sensing. Academic Press, 491-561. ISBN 9780128229422. (2023). doi:10.1016/B978-0-12-822942-2.00010-8.

21) Filip Wojciesz : Introduction to Quantum Computing with Q# and QDK. Quantum Science and Technology, Springer. ISBN:978-3-030-99379-5. (2022). doi:10.1007/978-3-030-993795.

22) Aniruddh Bharath : Quantum Computing With Qiskit. Conference on DRiP fall 2020. (2022). doi:10.13140/RG.2.2.29107.35360.

23) Ali Javadi-Abhari, et. Al: Quantum computing with Qiskit. (2024). arXiv:2405.08810.