Comprehensive Report

and

Project Synopsis

Quantum Computation

and

Quantum Information Processing

By,

Satish Bhambri

MASTERS

SOFTWARE ENGINEERING,

ARIZONA STATE UNIVERSITY.

PREFACE

This comprehensive report discusses the paradigm of Quantum Computation and Quantum Information processing, referring to the lectures and writings of Dr Umesh Vazirani on the subject of Quantum Computing and the book Quantum computation and Quantum Information and subsequently mentioned other research resources present in the bibliography. It presents a comprehensive study of the various topics in this research, the algorithm of quantum paradigm, which has been used to solve the problems which were deemed unsolvable in classical paradigm.

CCS Concepts: • **Computer systems organization → Quantum systems.**

KEYWORDS

Fourier Sampling, Simon's Algorithm, Reversible Computation, Simulating Classical qubits, Recursive Fourier Sampling, Quantum Fourier Sampling, Quantum factoring, Quantum Algorithms, Quantum Gates, Quantum Circuit, Continuous quantum states, Particle in a box, Observables, Expectation values, Unitary evolution, Quantization, Quantum computing.

CONTENTS

1 INTRODUCTION: Quantization

In last two decades, computing paradigm has seen several evolving phases but the one which has established altogether new roots in the computing paradigm is the quantization of classical paradigm, ie, the quantum computing. Our computing industry has been directed according to the Moore's law which predicts that the number of transistors on a Si chip doubles every 18 months to two years. But as the number of transistors are increasing, their size is decreasing only to enter the quantum realm where the classical physics' laws are no longer applicable. [1] [2] Hence, the quantum computing. Important features of Quantum mechanics which deviate from the classical paradigm are:

- Quantum mechanics' laws forbid the complete knowledge of the system's state; hence, a measurement only reveals a small amount of the information about the quantum state of the system.
- When we measure the state of a quantum system, this fundamental act disturbs the state of the system.
- Trajectories of quantum entities are not defined.
- Quantum mechanics is inherently probabilistic.
- Quantum entities behave both like waves and particles, depending on the conditions.

The dawn of quantum paradigm has led to evolution of algorithms which have solved upto great extent the problems which classical algorithms had tough time solving in the real time. For instance, the Bernstein – Vazirani algorithm, Simon's Algorithm, Factoring problema and Shor's Algorithms, Grover's search algorithm and many more, which we'll discuss in the context of this report.

2 SUPERPOSITION: Young's Double Slit Experiment

Light displays dual nature (wave – particle duality), behaving as a stream of particles or corpuscles (as Newton called them) in some situations and as an electromagnetic wave in some. Young's double slit experiment, performed in early 1800's, established the wave nature of light. This experiment includes a monochromatic light source, and an equidistant, thin and identical slits, as shown in the figure, and a detecting screen. [2] When the light source is turned on, we see interference patterns on the detector screen, thereby, the wave nature of the light is established.

But when we decrease down the intensity of the light to very low, which Young was unable to do, when this experiment was performed originally, such that we make sure that only single photon is released every second, then what we observe is in total contrast to the intuition. And we observe similar results if we take electrons, which intuitively we consider as particles. [3] [1] [2]

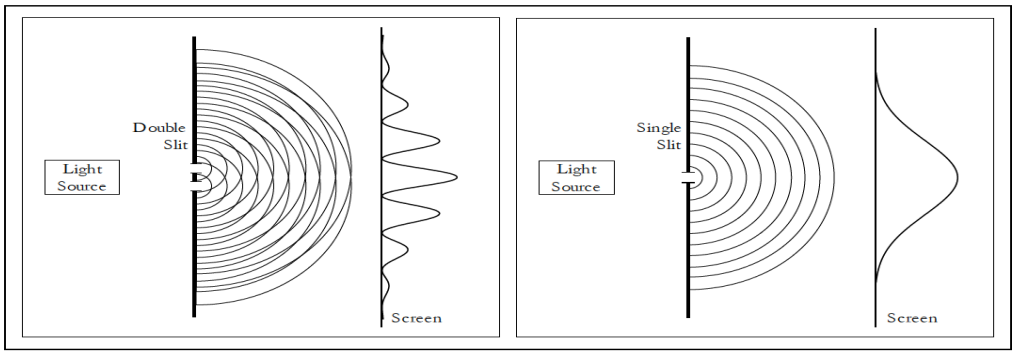
A brief insight into the experiment, consider a stream of bullets, instead of the light source. Now at the detector screen, we will observe a distribution pattern. If we close one of the slits, the bullets would only be detected behind the opened one and some deflected bullets in the

surrounding area. Hence, the normal distribution.

Consider a point y on the detector screen, such that P1(y) denotes the probability that the bullet lands at point y when only slit 1 is open, and correspondingly P2(y). Say, that P12(y) denotes the probability of bullet landing at point y when both the slits 1 and 2 are open.

$P12(y) = P1(y) + P2(y)$.

Considering, waves (for instance, water waves), then we see the interference pattern as shown in the figure below.



[2]

In this case, we observe dark patches where waves are out of sync and very bright patches where they are in sync and positively superimpose each other.

Calculations, in the case of waves involve the height of the waves or the Amplitudes.

$H12(y) = H1(y) + H2(y)$

Intensity at any point y would be given by the,

$I12(y) = (H12(y))^2$

Hence, the difference from the case of bullets. And hence, the interference pattern in case of waves.

Now, coming back to our point of decreasing the intensity of light such that only one photon passes through, we intuitively expect the nature of distribution to resemble the distribution pattern of the bullets. In this case, photodetectors on our detector screen would report the photon every second, and only one photodetector would do so every second. [2] [1]

Logically, one photon should go through one slit at a time, producing the bullet pattern, but in this case, we still observe the interference pattern. The explanation is that, since the

trajectories of a quantum system is not defined, hence photon goes through both the slits and interferes with itself. [3] [2] [1]

If we try to close one slit, then the interference pattern goes away and we get normal distribution. This reflects the measurement principle of the Quantum paradigm, wherein measuring the system alters the state of the system.

This experiment with the stream of electrons yield the same results following the following amplitude equations.

$A12(y) = A1(y) + A2(y)$

$P12(y) = |A12(y)|^2$.

Where, A12(y) represents the amplitude of the resultant wave after interference at any point y and P12(y) represents the Probability that the photon is detected at the point y, when both slits 1 and 2 are open.

We exploit this superposition nature of the Quantum paradigm to our advantage in the quantum algorithms.

3 AXIOMS, QUBITS AND KET NOTATION

Following are the basic axioms of Quantum Mechanics and hence the quantum computing [3]:

- **The Superposition principle**: This principle describes how a particle can be superimposed among multiple state at the same time. This is the most important step in any quantum algorithm, wherein we set up the desired superposition as required by the problem, before performing the measurement, which is the subsequent step.
- **The Measurement principle**: This principle describes how the measurement of a particle, changes it's state and the amount of information that we can access from a particle. Measuring a superimposed state in a quantum system, sets the state of that quantum system to the value which has been measured with some probability. Interesting part and the most important part about this measurement in our quantum algorithm is that we need not set up the whole apparatus again and again in order to yield the different outputs, rather give the same input again in the same setup and measure to obtain another set of output. We exploit this advantage to gain the different samples for instance in period finding and gain significant advantage over classical paradigm.
- **The Unitary evolution**: This axiom states the evolution of a quantum system in time. This aspect of Quantum paradigm is used to transform the superimposed states into our required basis states such that suitable measurement could be performed.
    - Our quantum gates are based on this principle, for instance, Hadamrad gate which performs a unitary transformation in the two dimensional complex vector space for a given qubit.
    - Unitary evolution forms integral step of the most efficient quantum algorithms like Grover's algorithm and Shor's factoring algorithm.

3.1 **The Superposition principle**:

Let's take a circuit or a system, such as hydrogen atom, with k different states. In this case, an electron of the hydrogen atom can be in one of the discrete set of energy levels, beginning from the ground state, the first excited state, the second excited state and so on. We denote the k different levels of our hydrogen atom or k different states as |0>, |1>, …, |k-1>, being the ground state, first excited state, .., $(k-1)^{th}$ excited state, respectively. Superposition principle tells us that the state of the electron is given by :

$|\psi> = \alpha 0|0> + \alpha 1|1> + \ldots. + \alpha k\text{-}1|k\text{-}1>$

Where ψ represents the wave function, describing the system, and α0, α1, αk-1 the normalized, complex coefficients complex coefficients such that $\Sigma_j|\alpha_i|^2 = 1$.

This notation, $|\psi> = \alpha 0|0> + \alpha 1|1> + \ldots. + \alpha k\text{-}1|k\text{-}1>$, is called the **Dirac's Ket notation** and the normalization on the complex amplitudes means that the state is a unit vector in a k dimensional complex vector space, known as, Hilbert space. [2] [1] [3].

We perform our next steps of Quantum Measurement and Unitary evolution on this Hilbert space itself, as we'll discuss further, to obtain the required logical circuit for our algorithms.

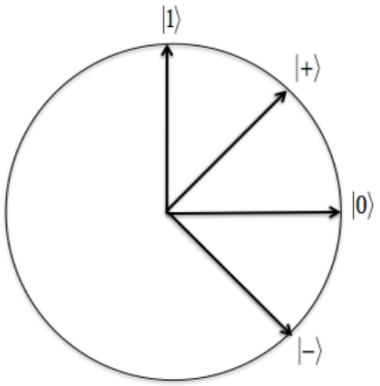


Figure 1.2: Representation of qubit states as vectors in a Hilbert space.

Mathematically, we can write mutually orthogonal vectors as:

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \quad |k-1\rangle = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{pmatrix}.$$

These k mutually orthogonal unit vectors, in k-dimensional complex vector space form the orthonormal basis for that state space, and are called the **standard basis**. Hence, given any two states $\alpha 0\ |0> + \alpha 1\ |1> + \bullet \bullet \bullet + \alpha k{-}1\ |k-1>$, and $\beta|0> + \beta|1> + \bullet \bullet \bullet + \beta|k-1>$, we can compute the inner product of these two vectors, which is $\Sigma\alpha^* j \beta j$. Hence, for orthogonal vectors, their inner

product needs to be zero, as the absolute value of the inner product is the cosine of the angle between these two vectors in Hilbert space. [2]

Consider a state of the system as given below and where the kets |k> form the basis as

$$|\psi\rangle = \sum_k a_k |k\rangle ,$$

$$|\psi\rangle = \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_{N-1} \end{pmatrix}$$

Then, the inner product of this state with itself is given by:

$$\langle \psi, \psi \rangle = \begin{pmatrix} a_0^* & a_1^* & \cdots & a_{N_1}^* \end{pmatrix} \cdot \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_{N-1} \end{pmatrix} = \sum_{k=0}^{N-1} a_k^* a_k = \sum_{k=0}^{N-1} |a_k|^2$$

Dirac introduced a better representation of this inner product, by defining the conjugate transpose of a ket as “bra”, and represented by [2] [1] :

$$\langle \psi | = |\psi\rangle^\dagger = \sum_k a_k^* \langle k| .$$

This object acts on the ket as a function and gives a number. Hence the inner product now can be calculated as :

$$\begin{aligned} \langle \psi | \psi \rangle &= \left(\sum_j a_j^* \langle j| \right) \left(\sum_k a_k |k\rangle \right) \\ &= \sum_{j,k} a_j^* a_k \langle j|k\rangle \\ &= \sum_{j,k} a_j^* a_k \delta_{jk} \\ &= \sum_k |a_k|^2 \end{aligned}$$

Now we can write the inner products of any two states as follows:

$$\langle\psi|\phi\rangle = \sum_{j,k} a_j^* b_k \langle j|k\rangle = \sum_k a_k^* b_k$$

$$|\phi\rangle = \sum_k b_k |k\rangle .$$

$$\langle\psi|\phi\rangle = \langle\phi|\psi\rangle^* \in \mathbb{C}$$

3.2 **The Measurement principle**:

This principle describes the extraction of the information from the superimposed world of an electron or any particle displaying superposition. A measurement of this k-state superimposed system yields one of the k possible outcomes with the probability of that outcome to be the square of the magnitude of complex coefficient, but it also alters the state of the system, such that the new state is exactly the outcome of the measurement, that means that if the outcome of the measurement is j, then after the measurement the qubit is in state |j>.

Hence, you can't collect any additional information following about the amplitudes following the measurement. Therefore, measurement is a probabilistic rule for projecting the state vector on the one of the vectors of the orthonormal basis [3] [2] [1].

Measurement of a set of qubits in a superimposed states yield a particular state of the system, but as we'll see later on, the measurement can be deferred, that is the other two aspects, setting up the superposition and evolution of a system in different basis states play larger role in the determination of the state of the system, particularly, we can say that unitary evolution gives a direction to the randomness of the output, it binds the output to be random in some particular basis, which can be determined using the measurement.

3.3 **Qubits**

Qubits are the quantum bits, quantum analog of the classical bits, and they form the basic building blocks having all fundamental quantum phenomenon. Qubits are basically, 2 state quantum systems. For instance, taking the hydrogen atom and considering only two of its states.

$$|\psi\rangle = \alpha||0\rangle + \beta\,|1\rangle \qquad \text{with} \quad \alpha, \beta \in \mathbb{C} \quad \text{and} \quad |\alpha|^2 + |\beta|^2 = 1.$$

The simplest measurement is in the standard basis, and measuring |ψ> in this {|0>, |1>} basis yields 0 with probability |α| squared, and 1 with probability |β| squared and this measurement alters the state of the system. Examples of Qubits could be atomic orbitals, photon polarization, and spin of the electrons.

4 Two Qubits and Entanglement

Let us consider a two-state quantum system consisting of two qubits, which is described by two hydrogen atoms, such that considering the system to be comprising of two electrons from these two atoms, wherein the electrons can either be in ground state for both atoms or excited state for both atoms or excited for first atom and in ground state for the second or vice versa. So basically, these four states are possible.

By superposition principle, quantum state of the aforementioned system can be represented by the following state vector and thus the system can be in any linear combination of these four classical states:

$$|\psi\rangle = \alpha_{00}|00\rangle + \alpha_{01}|01\rangle + \alpha_{10}|10\rangle + \alpha_{11}|11\rangle \quad [2]$$

The measurement of this 2-qubit system reveals only two bits of information with the probability as the square of the coefficients of that state. For example, say we measure the above system, the probability that we'll find both the qubits in the state 0(|00>) is $|\alpha_{00}|^2$. Then the system will fall into a state where both electrons will be in the ground state as per the measurement principle. [1] [2]

Considering the scenario where we measure the first qubit to be in 0 and not the second one, what and how can we determine the probability outcome for this case. Actually, the probability outcome is exactly the same as it would have been had we measured both the qubits. [2]

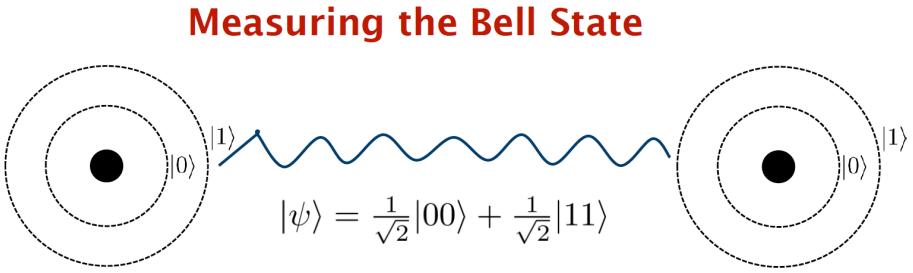
$\Pr\{1\text{st bit} = 0\} = \Pr\{00\} + \Pr\{01\} = |\alpha_{00}|^2 + |\alpha_{01}|^2$

Therefore, in a general sense we can say that given the state of first qubit to be $\alpha_0|0\rangle + \alpha_1|1\rangle$ and the state of the second qubit to be $\beta_0|0\rangle + \beta_1|1\rangle$, then the combined state of the two qubit system is given by $\alpha_0\beta_0|00\rangle + \alpha_0\beta_1|01\rangle + \alpha_1\beta_0|10\rangle + \alpha_1\beta_1|11\rangle$.

But can every state of two qubits be decomposed in this way? We have found some states in nature which cannot be. They are of the form $|\Phi^+\rangle = 1/\sqrt{2}\,(|00\rangle + |11\rangle)$. Such states are called the **Entangled states**, giving rise to the phenomenon of **Entanglement**.

When we have two entangled qubits, we cannot determine the state of each qubit independently. For instance, say, if the first and respectively, the second qubit of $|\Phi^+\rangle$ is measured then the outcome is 0 with probability 1/2 and 1 with probability 1/2. However, if the outcome is 0, then a measurement of the second qubit results in 0 with certainty. This is irrespective of the spatial separation between the two particles.

We use this property of the Quantum circuits to safely teleport information from one lab to another, without destroying the superposition by performing the actual measurement.

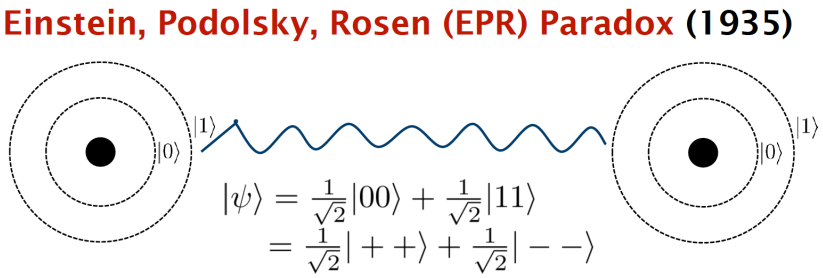


[2] [1]

5 EPR Paradox

Albert Einstein considered Quantum mechanics to be an incomplete theory and believed that randomness of quantum measurements reflected our lack of knowledge about additional degrees of freedom, or "Hidden variables", of the quantum system. He, along with Podolsky and Rosen worked on this theory in a paper in 1935, introducing the **Bell states** ($|\Phi^+\rangle = 1/\sqrt{2}$ ($|00\rangle + |11\rangle$)). For the Bell states, if we measure the first qubit in the bit basis, the other qubit is determined in the bit bases, irrespective of how far they are apart. [2] Considering an instance from this line of reasoning, assume that the qubits are very far apart, say one light second, and we measure the qubit 1 in the standard basis and half a second later we measure qubit 2 in the same basis; the two measurements must agree. But qubit 2 could not possibly know which basis was qubit 1 measured in until a complete second after we measure it because light itself takes one second to reach from qubit 1 to qubit 2. Both qubits couldn't have communicated any information in that time.

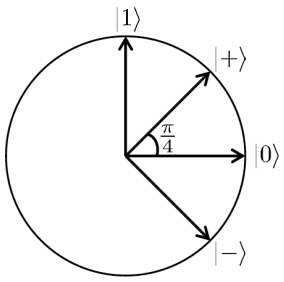
From the above findings, Einstein, Podolsky, and Rosen formulated the result that because qubit 2 cannot have any information about which basis qubit 1 was measured in, its state in both bit and sign bases is simultaneously determined, which is something that quantum mechanics does not allow. Hence, they suggested that quantum mechanics is an incomplete theory, and there is a more complete theory where "**God does not throw dice**." or a " local hidden variable theory" which describes the predictions of quantum mechanics, but without resorting to probabilistic outcomes.



6 Bit and Sign Bases

As discussed, we have a orthonormal basis represented by quantum analog of classical bits 0 and 1 as |0> and |1>, which is called the Bit Bases, but these are not only the possible bases which

can be used to represent any states. We can have infinitely many orthonormal bases, which can be used to represent a quantum state. Another important is the Sign basis which are obtained by rotating the Bit basis by an angle of 45 degrees on the geometric plane and represented by |+> and |->. [3] [1] [2]



[2]

7 **Uncertainty Principle**

Uncertainty principle given by Werner Heisenberg states that "One can never know with perfect accuracy both of those two important factors which determine the movement of one of the smallest particles- its position and its velocity." [2]

Quantum analog of this principle deploys the Bit and Sign basis, where the Bit basis corresponds to position and Sign basis corresponds to velocity/ momentum. So the principle boils down to the question of if we can know both bit and sign of a qubit simultaneously? Bit of a qubit can be |0> or |1>, and the Sign of the qubit can be |+> or |->. To quantify this, we define an entity Spread of a quantum state. Consider a quantum state being represented in Bit and Sign basis as follows: [2] [1] [3]

$$|\psi\rangle = \alpha_0|0\rangle + \alpha_1|1\rangle = \beta_0|+\rangle + \beta_1|-\rangle$$

Correspondingly, we define the spread in standard and sign basis, respectively as:

$$S(|\psi\rangle) = |\alpha_0| + |\alpha_1|$$

and,

$$\hat{S}(|\psi\rangle) = |\beta_0| + |\beta_1|$$

Therefore, the spread for |0> and |+>, in both Bit and Sign basis respectively could be calculated as follows :

$$S(|0\rangle) = 1 + 0 = 1 \qquad \hat{S}(|0\rangle) = \frac{1}{\sqrt{2}} + \frac{1}{\sqrt{2}} = \sqrt{2}$$

$$S(|+\rangle) = \frac{1}{\sqrt{2}} + \frac{1}{\sqrt{2}} = \sqrt{2} \qquad \hat{S}(|+\rangle) = 1 + 0 = 1$$

We have defined the spread this way because of the following reasoning. As per the aforementioned calculations, if we know the bit value perfectly, |0> or |1>, the spread is 1, in either case. But in the case that we don't know the bit value, say in case of |+>, that is we have the state plus, then alpha 0 and alpha 1 are both 1 over square root 2 and therefore, the spread is square root 2. Hence, the only way the spread can be small implying it to be 1, is if you know the bit perfectly. And the farther from 1 it is, the less certain you are about the bit value. Same is the scenario for |+> and |-> basis. [3] [1] [2]

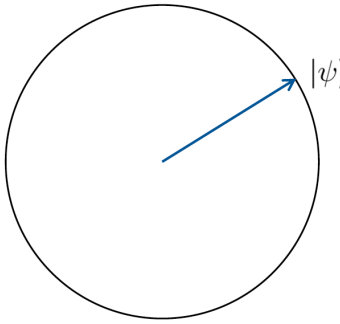
Thus, Uncertainty principle for bit and sign states is that if you look at the spread in the standard basis and multiply by the spread in the sign basis or any qubit, then this product is at least square root 2. Which means that both values cannot simultaneously be 1, at least one of them has to be square root of square root of 2.

Uncertainty principle for bit and sign: $S(|\psi\rangle)\hat{S}(|\psi\rangle) \geq \sqrt{2}$ for any $|\psi\rangle$.

Therefore, in a quantum circuit of an algorithm, we can be only sure about a qubit being in one basis states, we can't be sure about its any orthogonal state, which signifies that unless we perform a measurement, we cannot have any deterministic information about the qubits rather just the superimposed probabilistic amplitudes.

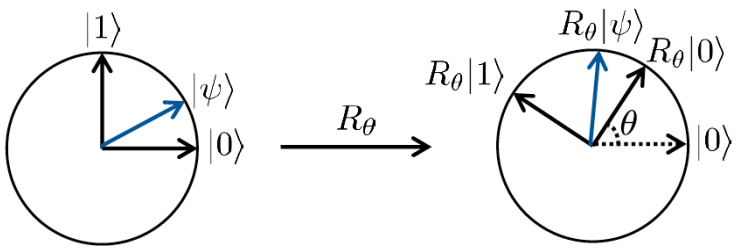
8 **Unitary Evolution**

Principle of Unitary evolution defines how a system evolve in time, by the rotation of the Hilbert space.



$$|\psi\rangle = \alpha_0|0\rangle + \cdots + \alpha_{k-1}|k-1\rangle = \begin{pmatrix} \alpha_0 \\ \vdots \\ \alpha_{k-1} \end{pmatrix} \in \mathbb{C}^k$$

The angles between the vectors are preserved while rotating, thereby its analogous to the rigid body rotation. This rotation is a **linear transformation** represented by the matrix. [2] [1]



$$R_\theta = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}$$

9 Quantum Gates

We discussed the Unitary evolution, which, geometrically is the rigid body rotation of the Hilbert space, thus resulting in the transformation of the quantum state vector such that the length of that vector remains constant during the transformation. We specify a Unitary transformation of the given vector in the Hilbert space by mapping the basis states |0> and |1>
to orthonormal states $|v_0\rangle = a|0\rangle + b|1\rangle$ and $|v_1\rangle = c|0\rangle + d|1\rangle$, a linear transformation on C^2(The complex vector space). [3]

$$U = \begin{pmatrix} a & c \\ b & d \end{pmatrix} \qquad U^\dagger = \begin{pmatrix} a^* & b^* \\ c^* & d^* \end{pmatrix}$$

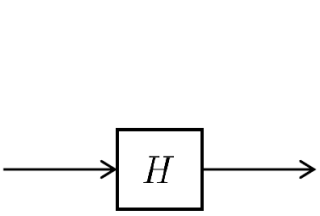
Where U and U perpendicular represent the transformation matrix and transpose of the transformation matrix respectively, and they satisfy the following relation:

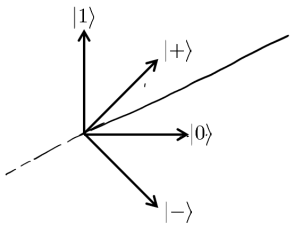
$$UU^\dagger = U^\dagger U = I.$$

Quantum gates are basically these unitary transformations on the qubits. Some of the prominent One qubit quantum gates are : [1] [2]

- **Hadamard gate**

Hadamard gate is the unitary transformation which is reflection around Pi/8 axis in the real plane.





$$H = \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix}$$

$$H|0\rangle = |+\rangle = \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$$

$$H|1\rangle = |-\rangle = \frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle$$

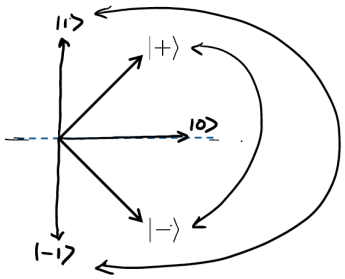
- **Rotation gate**

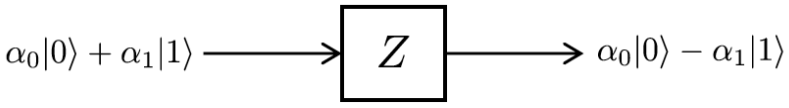
Rotation gate transforms the state by rotating the plane by an angle theta.

$$U = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}$$

- **Phase flip gate**

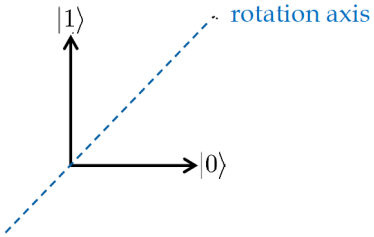
$$Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

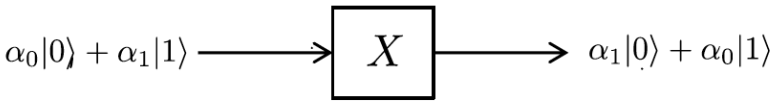




- **Bit flip/ NOT gate**

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

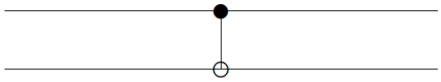




Basically, Phase flip gate is the NOT gate acting in the |+> = 1/√2(|0> + |1>) and |−> = 1/√2(|0> − |1>) basis. Hence, Z |+> = |−> and Z |−> = |+>. Evolution of a two qubit system is given on C^4 Hilbert space, given by 4 * 4 matrix and the four columns of U specify the four orthonormal vectors $|v_{00}>$, $|v_{01}>$, $|v_{10}>$ and $|v_{11}>$ that the basis states |00>, |01>, |10> and |11> are mapped to by U.

A basic two qubit gate is given by CNOT (controlled-not gate):

$$\text{CNOT} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$



First bit of the CNOT gate, represented by the upper bit in the diagram on the right side, is the control bit which controls the second bit, the target bit. The target bit flips only if the control bit is 1, if the control bit is 0 then target bit remains the same. [2] [1]

10 Quantum Circuit

Quantum circuit comprises of set of qubits being acted upon by gates and transformations, thereby exploiting the quantum properties discussed above. Any Unitary transformation on a quantum state can represented by a sequence of CNOT gate and single qubit gates. An important case to consider is the application of single qubit quantum gate to the first qubit in a two qubit system and checking the behavior of the second qubit.

For instance we apply Hadamard transformation to the state :

|ψ> = 1/2 |00> − i/√2 |01> + 1/√2 |11>.

The first qubit has been applied this unitary transformation and therefore, it yields the following result (The state of first qubit after the transformation):

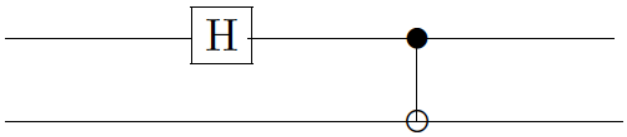
$|0\rangle \rightarrow 1/\sqrt{2}\ |0\rangle + 1/\sqrt{2}\ |1\rangle$ and $|1\rangle \rightarrow 1/\sqrt{2}\ |0\rangle - 1/\sqrt{2}\ |1\rangle$.

Hence the two qubit system is affected as :

$|\psi\rangle \rightarrow 1/2\sqrt{2}\ |00\rangle + 1/2\sqrt{2}\ |01\rangle - i/2\ |00\rangle + i/2\ |01\rangle + 1/2\ |10\rangle - 1/2\ |11\rangle$ resulting from the above three equations.

Therefore, $|\psi\rangle \rightarrow (1/2\sqrt{2} - i/2)\ |00\rangle + (1/2\sqrt{2} + i/2)\ |01\rangle + 1/2\ |10\rangle - 1/2\ |11\rangle$.

Now this can be used to design a very important quantum circuit, which can generate the **Bell states** ($|\Phi^+\rangle = 1/\sqrt{2}\ (|00\rangle + |11\rangle)$), the one responsible for entanglement. This quantum circuit consists of a Hadamard gate followed by a CNOT gate, and can be represented as follows:



First qubit is transformed using the Hadamard unitary transformation as did in the example above, then this transformed qubit is entangled with the other qubit using CNOT gate.

Consider the input to be $|0\rangle$ and $|0\rangle$, wherein the first qubit is subjected to Hadamard transformation and is changed to the state $\mathbf{1/\sqrt{2}\ (|0\rangle + |1\rangle) \otimes\ |0\rangle = 1/\sqrt{2}\ |00\rangle + 1/\sqrt{2}\ |10\rangle}$.

This state $\mathbf{1/\sqrt{2}\ |00\rangle + 1/\sqrt{2}\ |10\rangle}$ is then subjected to the CNOT gate which flips the second bit of the second qubit, as it's control bit is 1. Hence the state becomes $\mathbf{1/\sqrt{2}\ |00\rangle + 1/\sqrt{2}\ |11\rangle}$, which is a Bell state. [3] [1] [4] [2]

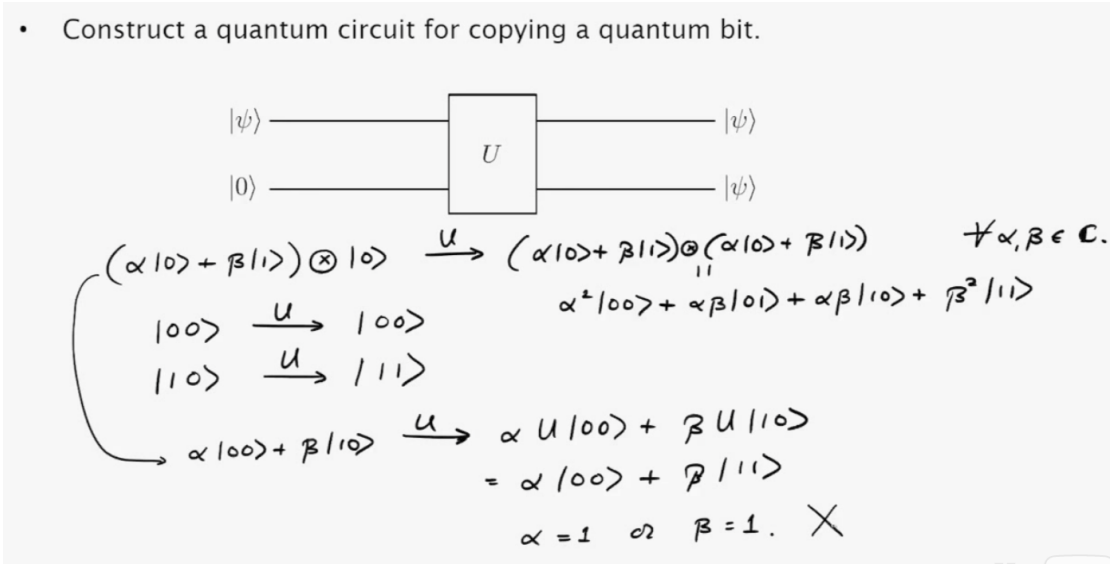
This is a primitive Quantum circuit, which can be used to generate the Bell states, and it also explains the implementation of a CNOT and Hadamard gates which are used widely in the quantum algorithms.

11 No Cloning Theorem

A very important case in the designing of the quantum algorithms is that of transferring the information from one place to another by making a copy of any particular quantum state. No cloning theorem delves into this realm of quantum computation. It caters to the very important question of if it is feasible to make a copy of any given quantum state $|\varphi\rangle = \alpha_0\ |0\rangle + \alpha_1\ |1\rangle$, that is create a state such that : $|\varphi\rangle \otimes |\varphi\rangle = (\alpha_0\ |0\rangle + \alpha_1\ |1\rangle) \otimes (\alpha_0\ |0\rangle + \alpha_1\ |1\rangle)$. Another way of asking this question is if it is possible to start with two qubits in state $|\varphi\rangle \otimes |0\rangle$ and transform them to the state $|\varphi\rangle \otimes |\varphi\rangle$?

By the third postulate of the Quantum mechanics, we should have a unitary transformation such that $U\ |\varphi\rangle \otimes |0\rangle = |\varphi\rangle \otimes |\varphi\rangle$. But this theorem proves that no such unitary transformation is possible, hence this operation is forbidden. [2] [1]

- Construct a quantum circuit for copying a quantum bit.



12 **Superdense Coding and Quantum Teleportation**

Consider Alice and Bob, connected by a communication channel which is capable of transferring the qubits. So in transferring quantum information from one place to another, another important aspect is to establish how many classical bits can Alice transmit to the Bob, in a message consisting of single qubit.

It is observed that if Alice and Bob share the quantum state which is entangled, and is a Bell state then, **Alice can send 2 classical bits by transmitting just one qubit over the channel**. [1] [2]

Let's consider the Proof.

Say, they both share the state $|\Phi^+\rangle = 1/\sqrt{2}\ (|00\rangle + |11\rangle)$, and that by applying suitable gate to her qubit, Alice can transform this shared state to any of the four Bell basis states $|\Phi^+\rangle$, $|\Phi^-\rangle$, $|\Psi^+\rangle$, $|\Psi^-\rangle$.

$$|\Phi^+\rangle = \frac{1}{\sqrt{2}}(|0\rangle_A \otimes |0\rangle_B + |1\rangle_A \otimes |1\rangle_B)$$

$$|\Phi^-\rangle = \frac{1}{\sqrt{2}}(|0\rangle_A \otimes |0\rangle_B - |1\rangle_A \otimes |1\rangle_B)$$

$$|\Psi^+\rangle = \frac{1}{\sqrt{2}}(|0\rangle_A \otimes |1\rangle_B + |1\rangle_A \otimes |0\rangle_B)$$

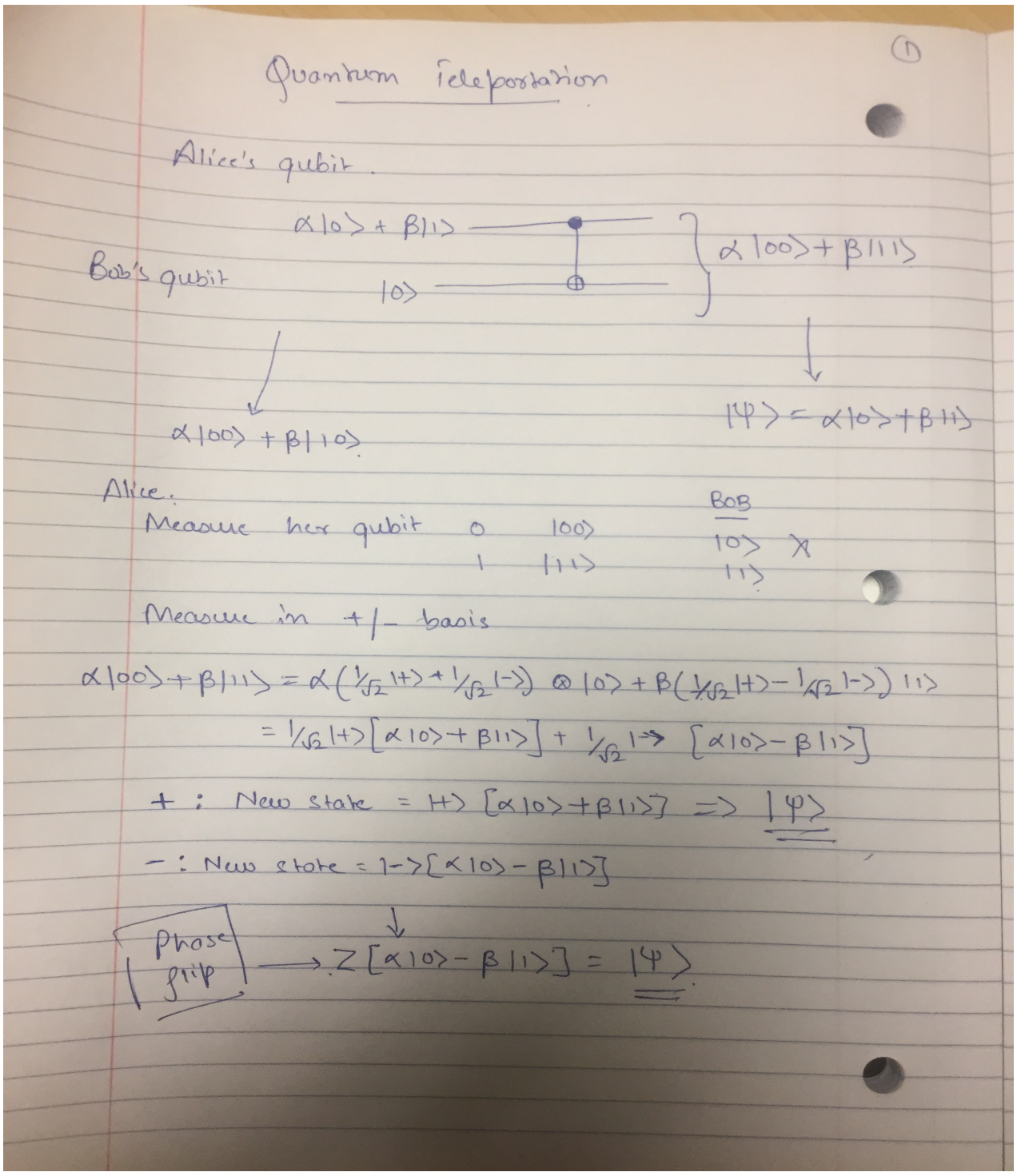
$$|\Psi^-\rangle = \frac{1}{\sqrt{2}}(|0\rangle_A \otimes |1\rangle_B - |1\rangle_A \otimes |0\rangle_B).$$

Now having got the Bell basis state, which is possible using two cases, first one being in case you have a CNOT gate between Alice and Bob as shown in the Notes image 1. When when she measures in the standard basis, she gets either 0 or 1, which if she conveys to Bob as a message,

he won’t be able to get the required quantum state. Hence, what she does is measure in the Sign basis states, wherein as per the calculations, she either gets |+> or |->. If she gets a |+> state, then she conveys that to bob using say a bit 0, which means Bob has already received the required quantum state and need not do anything. In case her measurement result is |->, then she conveys the bit 1, which means Bob needs to apply the phase flip gate in order to receive the required quantum state.

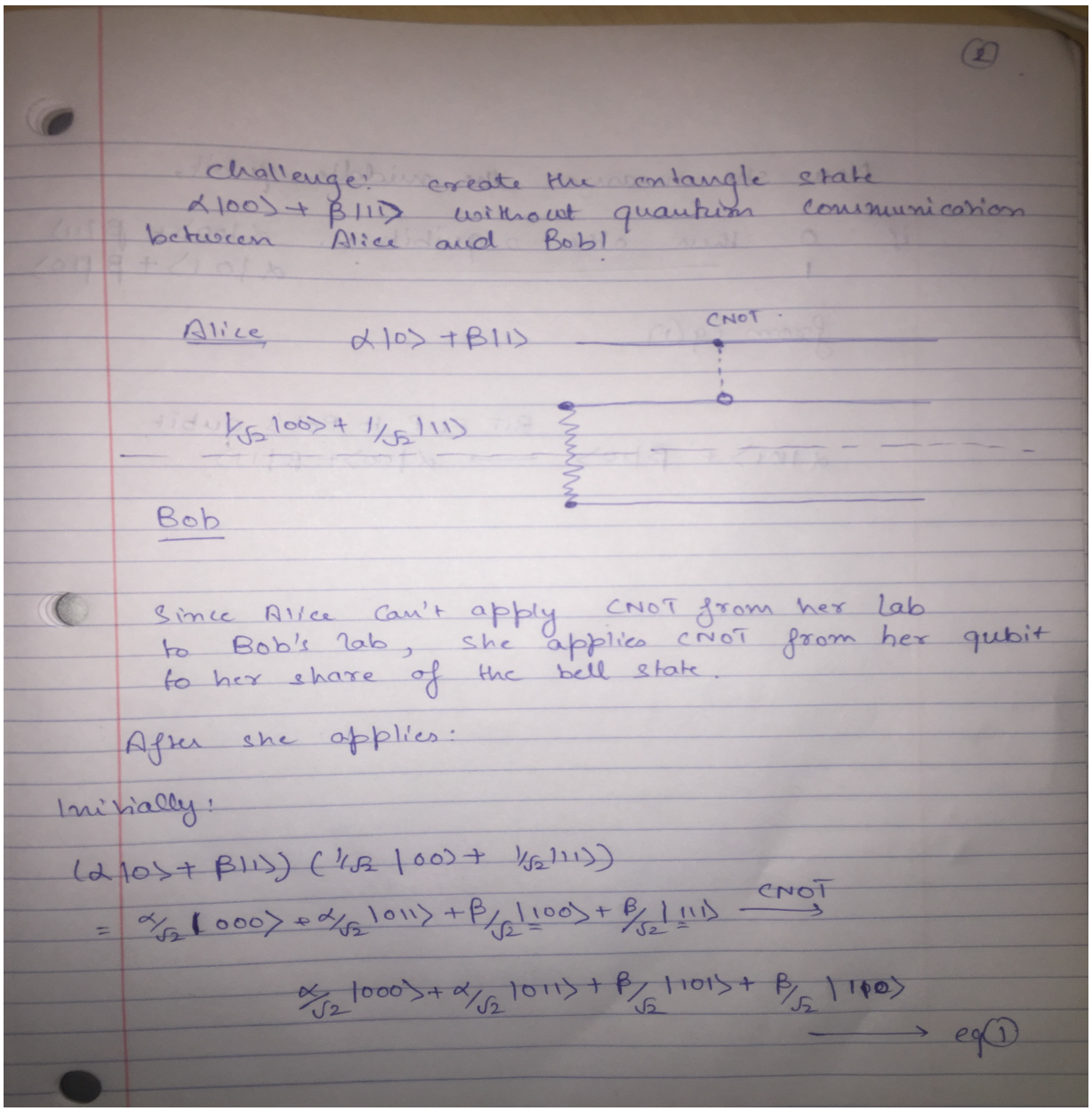
This part generalizes the fact of producing a same quantum state somewhere else in the circuit if we have it somewhere else, and at the same time not destroying the superposition.
The conservation of superposition is of crucial importance, because, else it renders the qubit useless afterwards, once we have measured the state, since our system would drop to that state and recovery of the previous state might not be possible.

Interesting part is such a state transfer happens with the minimal communication as we will see further.

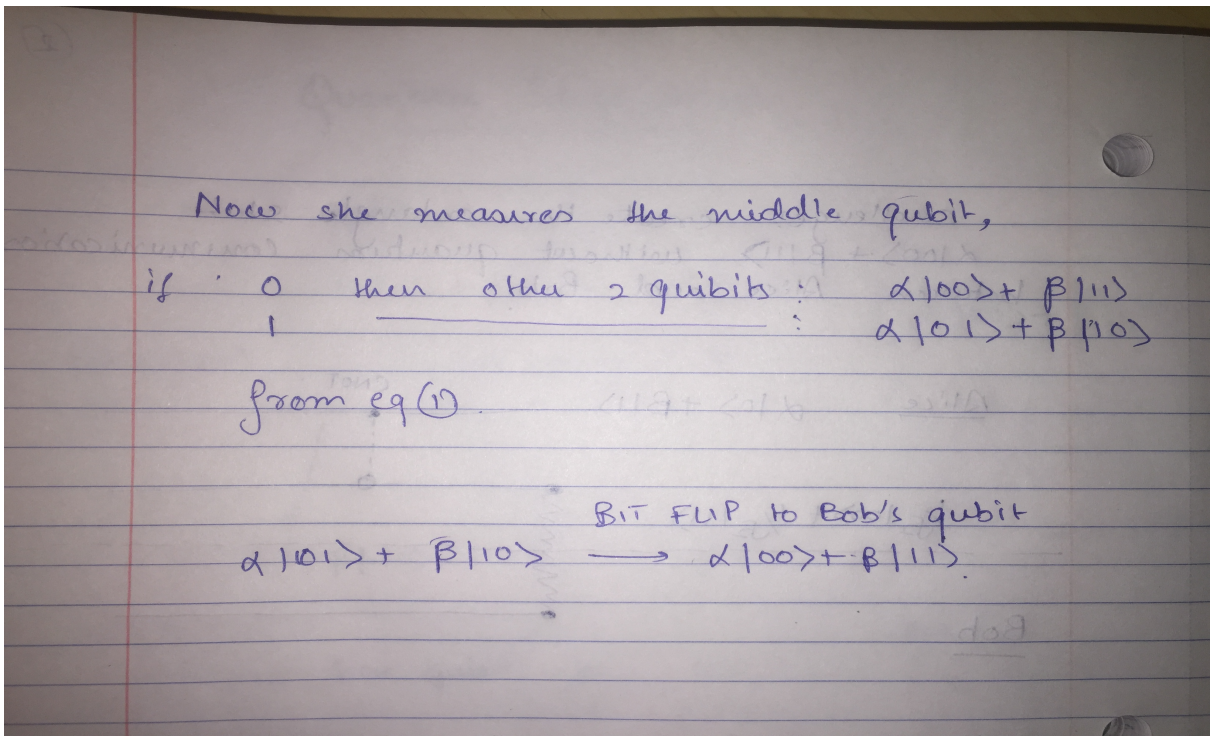


NOTES IMAGE 1

Actual challenge happens when Alice and Bob are very far apart and we can't have any shared CNOT gate between them. In such a case as shown in the Notes image 2, We consider 3 lines of communication being shared between Alice and Bob. First one is the one having Alice's qubit, middle one is shared and contains the shared Bell state for Alice, while third one represents the Bob's share of Bell state. [2] [1]

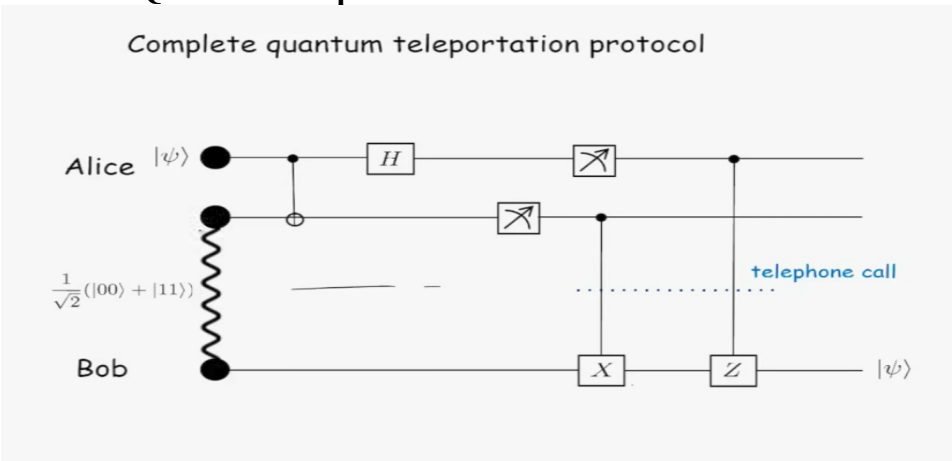


NOTES IMAGE 2



So Alice applies CNOT to her part of the lines and does the computation as shown in the notes. So, if she receives the qubit 0, she conveys 0 to Bob and Bob needs not do anything but has his required qubit. Whereas if she receives bit 1, then Bob needs to do a Bit flip for the required quantum state.

Hence the **Quantum Teleportation**.

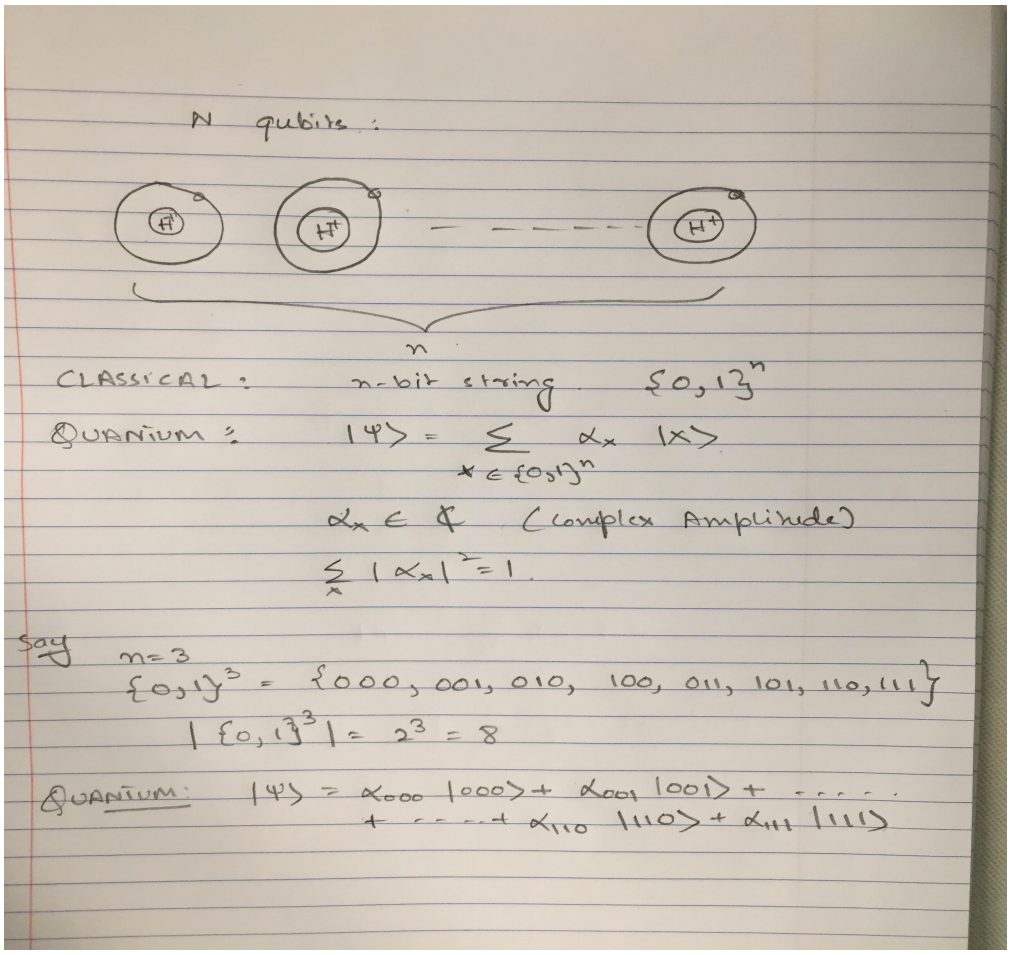


Notice that measuring in the sign basis is same as applying the Hadamard transform and measuring it in the standard basis.

13 **Quantum Algorithms**

13.1 **n qubit system**

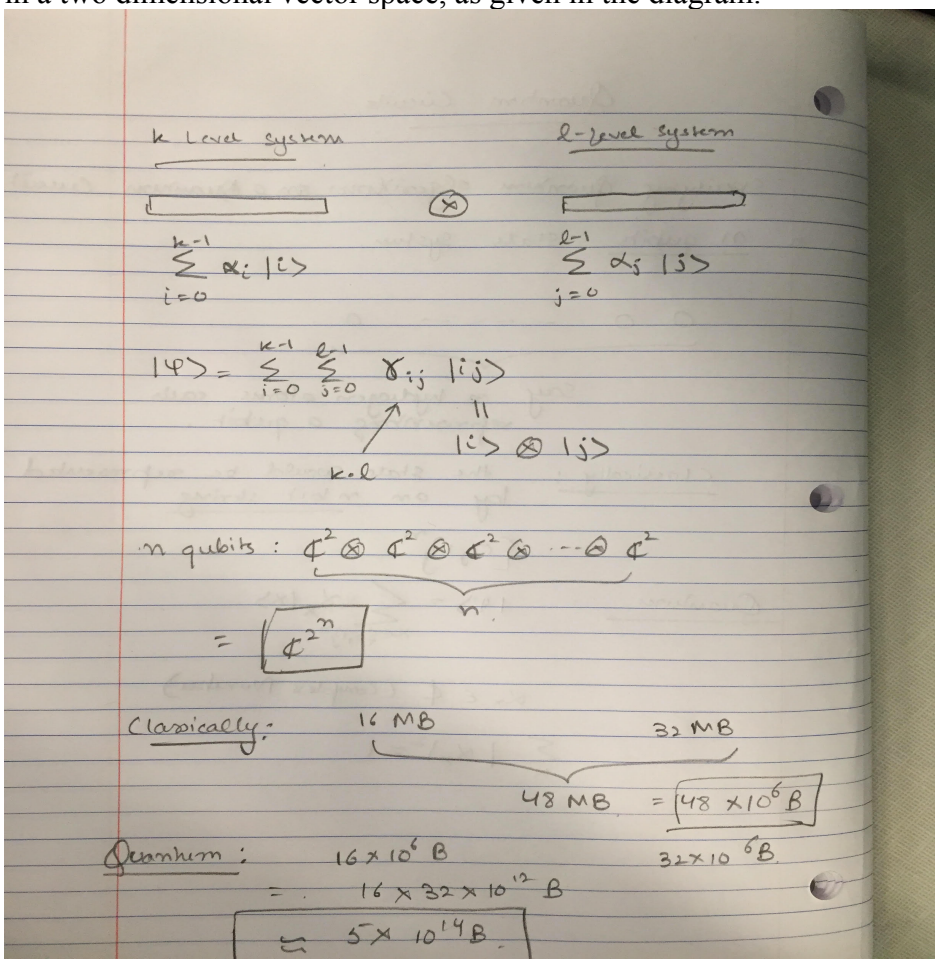
Now here, focus on how we specify a quantum algorithm in terms of a quantum circuit, considering that quantum circuit acts on a system of n qubits, what the state of an n qubit system looks like. Considering a qubit as the state of an electron in a hydrogen atom, say that we have many such hydrogen atoms such that they form a system of n qubits. In a classical paradigm, we represent this system using n bits, hence, the state of such a system is given by an n-bit string. In Quantum realm, by the superposition principle, the state is a superposition of all these classical possibilities, psi, is a superposition over all n-bit strings with probability amplitude alpha sub x, alpha sub x being a complex number. Consider the example given in the diagram.



We can see that exponential function grows extremely fast such that even for moderate values of n, say a few hundred, 2 to the power of n is already larger than the number of particles in the visible universe, or even the age of the universe in femtoseconds. Consider the following scenario to understand the exponential nature of the superposition, where we have two quantum systems, one a k level one and the other l level one.

These system states are written as a superposition of k different states (0 through k-1) and l different basis states respectively. Considering these two as a composite system. We get the state of this general system, which consists of this composite of these two different subsystems by taking tensor products as given in the diagram.

So we need k parameters, k complex numbers, to specify if we only had the first system and l parameters to specify only the state of the second system, but putting these systems together we need k*l parameters to specify the state of that system. Consider that we have a computer with 32 megabytes of memory and another one also with 32 megabytes. Combining these two systems, classically, we have 64 MB memory but Quantum mechanically, we have 32 * 32 MB and that is what happens when we take a system of n qubits it is that the state of each of them sits in a two dimensional vector space, as given in the diagram.



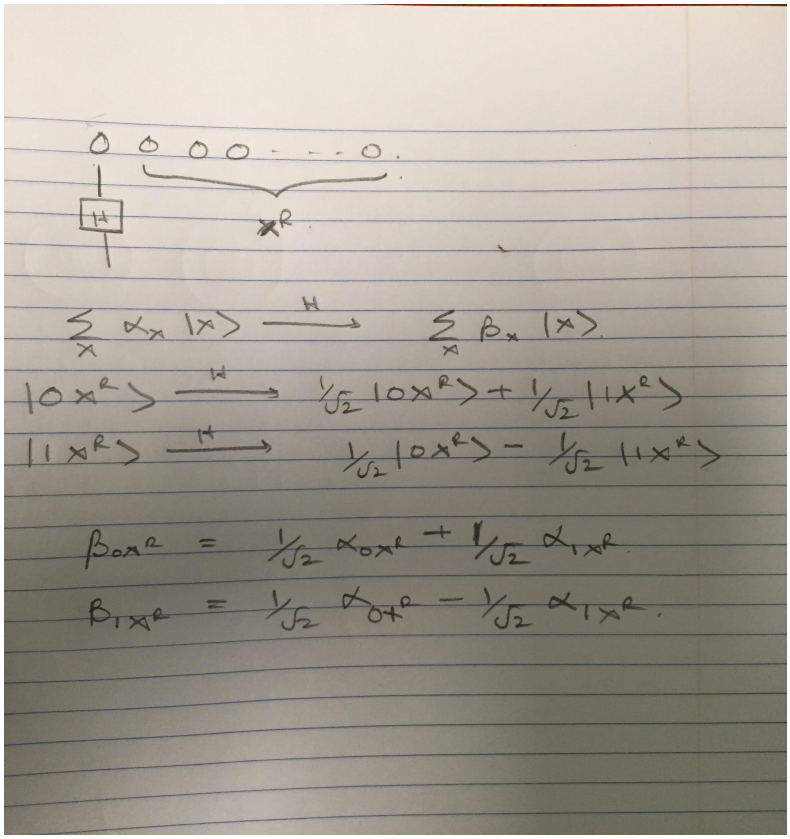
Now to implement a quantum computer, we need to effectively manipulate this exponentially large vector space and corresponding complex numbers. The question we next address is if we can manipulate all these exponentially many amplitudes efficiently and measure the results because this is what the potential for quantum algorithms is going to rely on.

13.2 **Manipulating n qubits**

To manipulate the aforementioned amount of data, we need to perform some kind of a quantum gate on atleast one a pair or on one of these qubits and we see that by doing that, all the exponentially many amplitudes get updated simultaneously. For example, as given in the diagram, we have our n qubits here. We perform the Hadamard transformation on the first qubit,

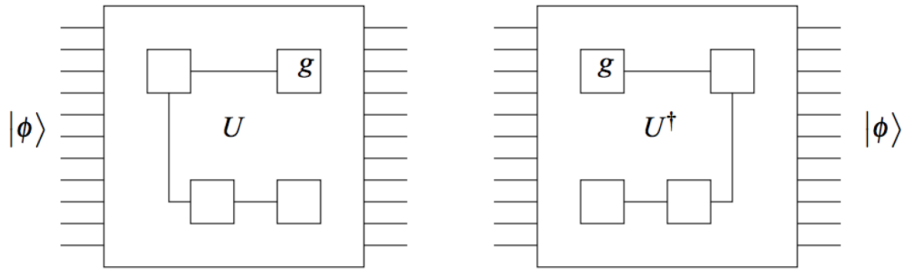
leave the remaining qubits as such. We are interested in knowing the state of the system after the transformation. Denoting the remaining qubits by X^R.
Such transformations are extremely important with respect to our quantum algorithms as we can manipulate the entire circuit by manipulating just few qubits.

Hadamard gate transforms a |0> to $1/2^{1/2}$ |0> and $1/2^{1/2}$ |1>, correspondingly |1>. Now the superimposed amplitudes after the transformation given by Beta, are changed as in the diagram. Logically, we had an exponential superposition to start with, and even if n was as small as a few hundred or a thousand, 2 to the n is much larger than the number of particles in the visible universe. An intriguing fact is to determine where nature stores such a large amount of information.



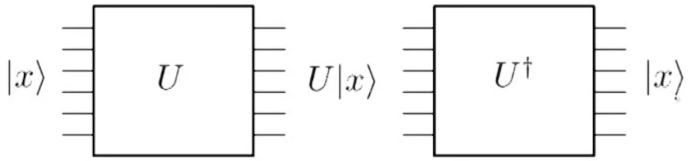
With such large amount of data present, if we do just a slight change, the underneath computation that takes part to change the amplitudes of the resulting superposition represents enormous computation that nature is carrying out, and quantum computation is trying to delve into that. But then finally, when we measure, we only get very limited access to this information and Hence, quantum algorithms is the art of making use of these resources that
quantum mechanics gives us extravagant resources with some degree of control, but very limited access, and to use those to solve a difficult computational problem.

14.0 **Reversible Computation**



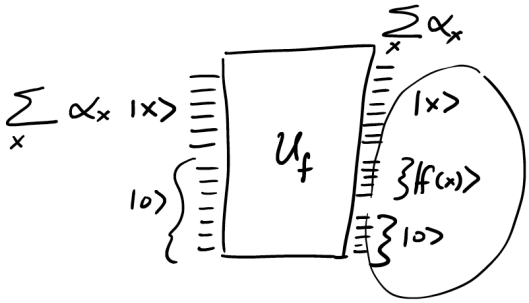
A quantum circuit acting on n qubits is described by an $2^n * 2^n$ unitary operator U. Since U is unitary, $UU^\dagger = U^\dagger U = I$. This implies that each quantum circuit has an inverse circuit which is the mirror image of the original circuit and which carries out the inverse operator $U^\dagger$.

Consider a function f(x) which maps n bits to m bits. We can create a quantum circuit describing that function, but that circuit would involve quantum gates, which are basically unitary transformations. Since, these transformations are unitary, they are reversible.

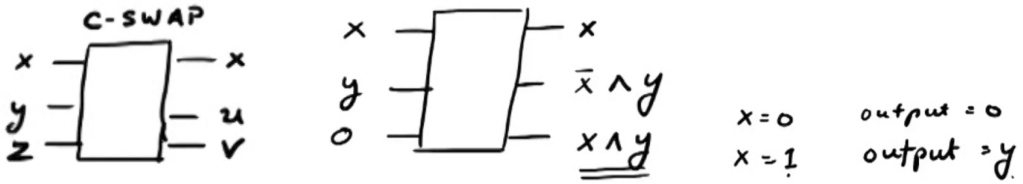


In this case, the dilemma that arises is that say m = 1, then after we have got the output, we have one bit of information, rest n – 1 bit information has been lost. How to recover that for computing to be reversible? We have this loss of information frequently in classical circuits. This is very important concept with regards to the quantum algorithms and quantum computation as a paradigm has evolved keeping this in mind.

So in quantum circuit we take a bunch of extra |0> as a work space, which produces in the output x, f(x), and remaining |0>.

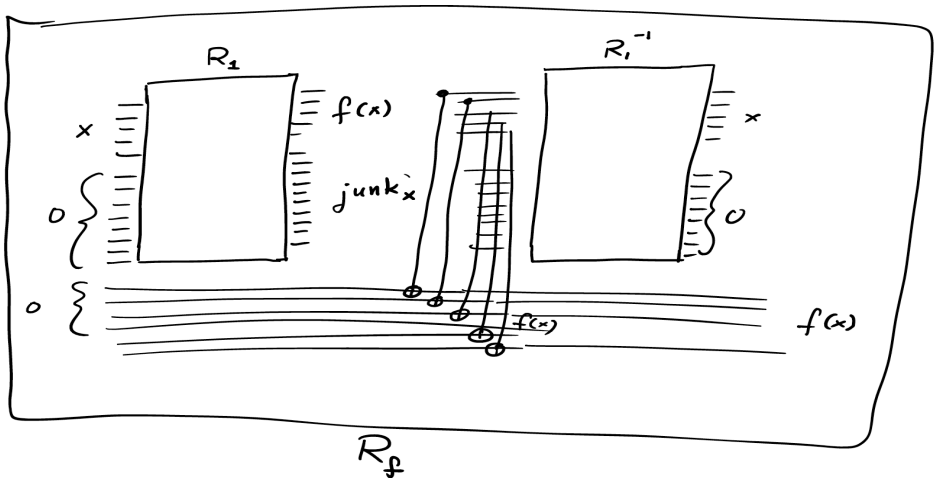


All the gates mentioned are universal, and reversible except AND. Simulating AND gate in a reversible manner is important to form the quantum circuits as C-SWAP gate. In this gate, as shown in the figure below, if x is equal to 1, then swap y and z; if x equal to 0, then u equal to y and v equal to z, else if x equal to 1, then u equal to z and v equal to y. We use this to create a reversible AND gate.

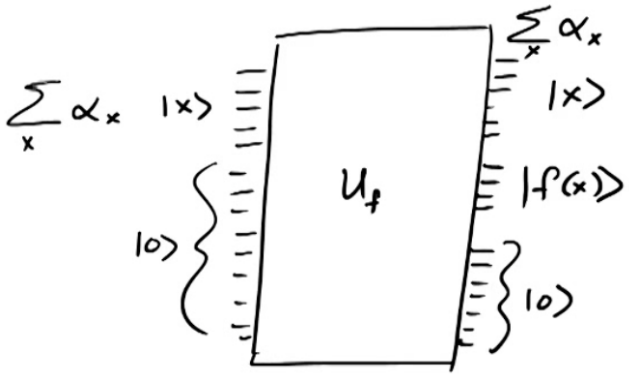


Hence, C-SWAP is used to simulate a reversible AND gate and C-SWAP is reversible in itself. Therefore, any Quantum circuit can be simulated reversibly in NOT, C-SWAP and CNOT gates.

A primitive reversible circuit looks like something this:



Notice that we add extra zeroes to keep a copy of f(x), because in the reversible process, to remove the junk(which we do to avoid unnecessary interference), we also remove the f(x), hence we need extra qubits to preserve the output in the reversible circuit.



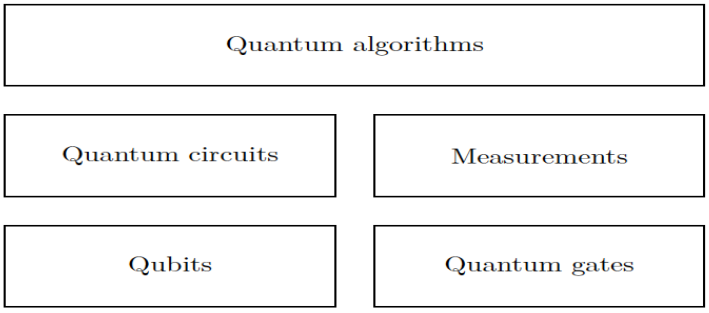
Quantum Computation and Quantum Algorithms : Project

Introduction

In this project we'll start with the Analysis of the quantum algorithms and continue with the simulation of a simple quantum circuit model, which would implement a basic qubit behavior and simple single qubit and two qubit gates. We'll create a GUI for testing the each of the Quantum circuit model, Shor's algorithm, Grover's Algorithm and Duetsch - Josza Algorithm.

Generally, there are four steps involved in quantum algorithms. Input qubits are initialised into some classical start state; the system is put into some superposition state; the superposition state is acted upon via unitary operations; some measurement of the system is taken, providing a classical output state. A brief description of these algorithms are given below.

Quantum Circuit model



A bit can represents two states, termed 0 and 1thereby, allowing us to store one piece of information: a yes or no (a Boolean value). Whereas a quantum bit can be described in terms of classical bit as in the figure.

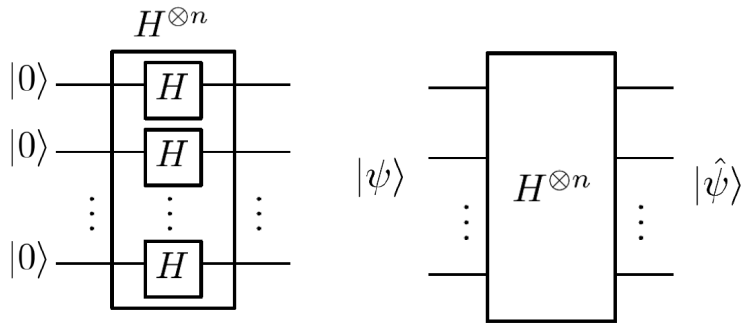
We'll use jQuantum set of library functions to describe and mimic the behavior of quantum bits and define functions to implement the corresponding gates.

The end user would be able to create his own circuit by setting the value of the qubits and applying the corresponding gate transformations to it.

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad \text{or} \quad |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

Quantum Fourier Sampling and QFT:

This is used to set up the stage for a quantum algorithm to function on, or it is the first step by which we create interesting superposition, to exploit the exponential power of Quantum circuit.



It gives us superposition in the form:

$$|u\rangle \longrightarrow \sum_{x \in \{0,1\}^n} \frac{(-1)^{u \cdot x}}{2^{n/2}} |x\rangle$$

where $u = u_1 u_2 \cdots u_n$

where u*x = u1*x1 + u2*x2 + .. +uN*xN. This Fourier sampling forms the basics building block in quantum algorithms, for **it gives us a state which can't be simulated classically.**

Parity Problem:

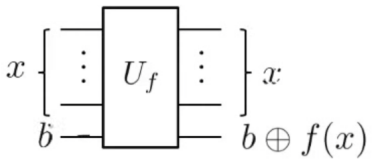
We have a function f(x) which maps n bits to a single bit as a black box and that f(x) = u*x for some u in $\{0,1\}^n$, where u*x = u1*x1 + u2*x2 + .. +uN*xN (mod 2). This problem looks for finding u with minimum queries.

Classically, we can approach in a brute force way, putting in the values as x = 1000 (say input is a 4 digit string), then sequentially, 0100, 0010 and so on, correspondingly calculating u1, u2, and so on. For an n digit string, we'll need n queries, as u is an n bit string.

$x = 100\cdots0$ $\quad f(x) = u_1$
$x = 010\cdots0$ $\quad f(x) = u_2$
$\vdots$
$x = 0\cdots01$ $\quad f(x) = u_n$

} n queries

Using a quantum circuit, we can perform this using only a constant number of queries, using Fourier sampling. Quantum circuit would look like this :



Bernstein – Vazirani Algorithm

This algorithm solves the Parity problem. It involves two steps :

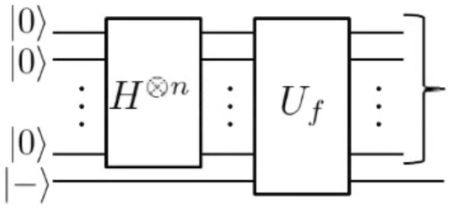
1. Setting up the following required superposition:

   $\frac{1}{2^{n/2}} \sum_x (-1)^{f(x)} |x\rangle$

   where f(x) here is u*x.

2. Performing Fourier Sampling to find u.

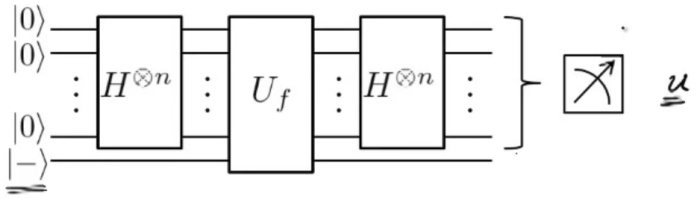
   This is because the superposition we have set up is nothing but the inverse of the set of Hadamard transforms applied on the |u> and which is a reversible gate. Hence, when we provide this superposition to Fourier Transform circuit, it gives u as an output. The important aspect here is to set up the superposition, for which we use the following circuit:



Important thing to notice is that instead of setting the answer qubit to |0>, we set it as |->.

Now, if f(x) = 0, the answer qubit remains the same, of it is 1, it flips the answer qubit to -|->,

This negative sign can be taken as a part of the superposition, and hence we get the desired superposition. Hence the final circuit being:



We can make this algorithm faster by using recursion which yield less number of queries but more expensive queries. This is done by **Recursive Fourier Sampling.**

Classically, T(n) > nT(n/2) + n, resulting in the order of:

$$T(n) = \Omega(n^{\log n})$$

Quantum Computationally, T(n) > nT(n/2) + O(n), resulting in the order of n(log n), which is just the time complexity of Merge sort. Hence we get exponential speedup.

Simon's Algorithm

We have a function f(x) which maps n bit strings to n bit strings, $f : \{0,1\}^n \rightarrow \{0,1\}^n$ and this function has a secret n bit string s such that

$$f(x) = f(x \oplus s)$$

We need to find this secret string s. Here, this addition is basically, bitwise addition mod 2. For instance, say if n = 3 and s = 101, and f(x) is defined randomly as:

x	f(x)
000	000
001	010
010	001
011	100
100	010
101	000
110	100
111	001

So classically when we try to find the structure of this function, we look for a collision, and it takes square root of N to find a collision, ie, $2^{n/2}$. Quantum computationally we can solve this problem in polynomial time.

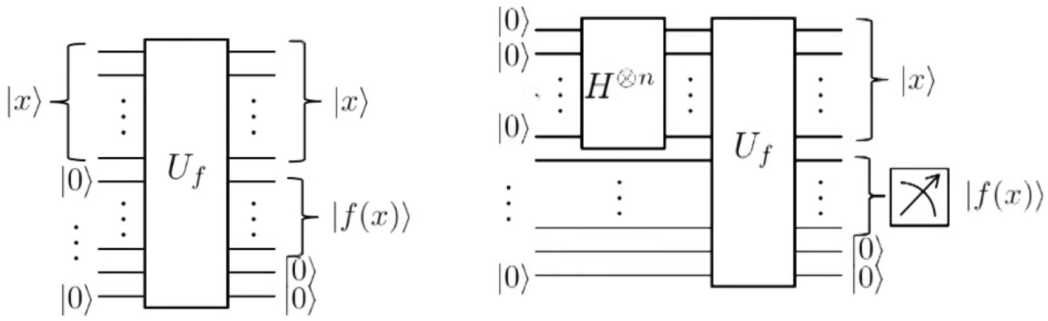
1. We set up a desired superposition as

   $$\frac{1}{\sqrt{2}}|r\rangle + \frac{1}{\sqrt{2}}|r \oplus s\rangle$$

   where s is the required secret string and r is a random n bit string.
2. We do the Fourier sampling to get a random y : y*s = 0(mod 2)

   Where y*s = y1s1+y2s2+..+yNsN = 0 (mod 2).
3. We repeat steps n-1 times to generate n-1 linear equations in s.
4. We solve these linear equations to get s.

To set up the required superposition, that is the first step, we use the following circuit:



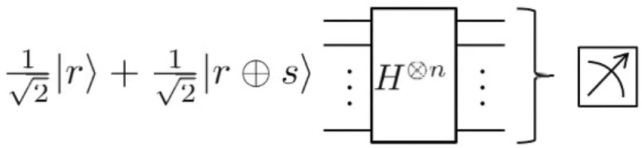
In the second circuit, after the H transform, we get a random superposition which is then manipulated by the function and hence we get f(x) here, which we measure. Now when we measure f(x) here, the superposition of |x> collapses to the values which yield f(x). But we know that there are two values which yield same f(x), ie, for a randomly measured f(r), we get:

$$a = f(r) = f(r \oplus s).$$

and hence, superposition of |x> reduces to the superposition:

$$\frac{1}{\sqrt{2}}|r\rangle + \frac{1}{\sqrt{2}}|r \oplus s\rangle$$

Then we do the next step, that is the Fourier sampling,



and we get some superposition in terms of beta, $\sum_y \beta_y |y\rangle$. Now here two cases are possible, such that y*s = 0(mod 2) in which case our beta would be $1/2^{n-1/2}$,

or y*s = 1(mod 2), in which case our beta = 0, hence they won't be visible. Hence, our beta evolves something like this :

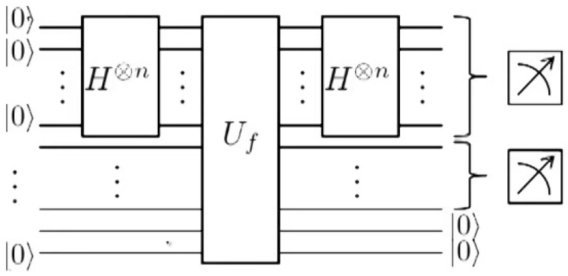
$$P_y = \frac{1}{\sqrt{2}} \frac{(-1)^{r\cdot y}}{2^{n/2}} + \frac{1}{\sqrt{2}} \frac{(-1)^{(r\oplus s)\cdot y}}{2^{n/2}}$$

$$= \frac{(-1)^{r\cdot y}}{2^{\frac{n+1}{2}}} \left[1 + (-1)^{s\cdot y} \right]$$

Now we sample and get n-1 linear equations in s as y1s1 + y2s2+ … + yNsN = 0 (mod 2), n-1 times.

$$\begin{aligned}
&y_1^{(1)} s_1 + y_2^{(1)} s_2 + \cdots + y_n^{(1)} s_n = 0 \\
&y_1^{(2)} s_2 + y_2^{(2)} s_2 + \cdots + y_n^{(2)} s_n = 0 \\
&\vdots \\
&y_1^{(n-1)} s_1 + y_2^{(n-1)} s_2 + \cdots + y_n^{(n-1)} s_n = 0
\end{aligned}$$

Hence the complete circuit would be :



Shor's Algorithm

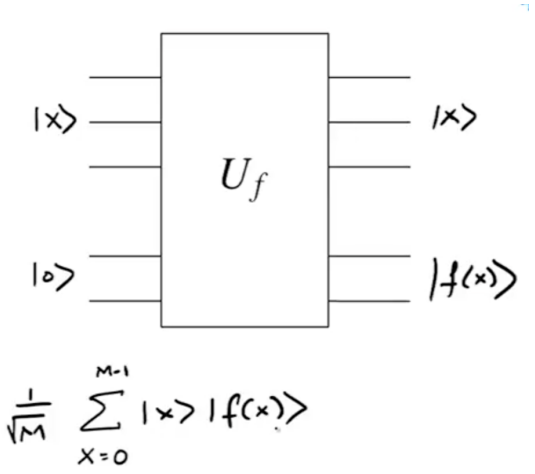
"The problem of distinguishing prime numbers from composites, and of resolving composite numbers into their prime factors, is one of the most important and useful in all of arithmetic... The dignity of science seems to demand that every aid to the solution of such an elegant and celebrated problem be zealously cultivated" — Carl Gauss

This algorithm involves **Period finding**. Let's consider a function f(x) which maps a set of numbers 0,1 ..., M-1 to a set S, such that we have for all x, f(x) = f(x+r). So basically, f is periodic with period r. If we have a domain of 100 numbers, ie, M = 100, and r = 5 then we have M/r = 20.

We make two assumptions here:

1. f(x) is one to one function.
2. M is divisible by r.
3. $M/r >> r \Rightarrow M >> r^2$.

Square root of r inputs are sufficient to find a collision in f(x) (also known as the Birthday Paradox).



$$\frac{1}{\sqrt{M}} \sum_{x=0}^{M-1} |x\rangle |f(x)\rangle$$

In Shor's algorithm, we carry out Fourier sampling, because of its properties of if we shift the superposition, the output of Fourier Sampling does not change. We do this so that our first non-zero value of the periodic function is not at any random place but at zero.

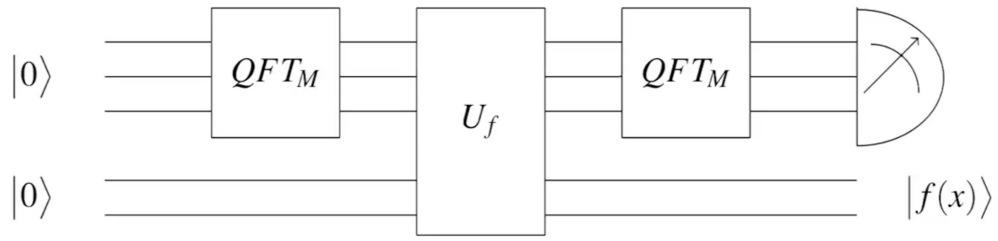
This yields us a graph where all non-zero amplitudes are multiples of the period, and gives us the superposition:

$$\frac{r=5}{\sqrt{\frac{r}{M}}} \sum_{j=0}^{\frac{M=100}{\frac{M}{r}-1}} |j\,r\rangle$$

Hence, the output we get is the random multiple of M/r. The example we took had M/r as 20, hence, in this case we'll get a random multiple of 20. Therefore, when we run this algorithm, the output may be 60 for the first time and probably 80 for the second time.

Then we calculate the gcd(Greatest Common Divisor) of the two outputs, we get 20. Now we have M/r and M, we calculate the period r as M/(M/r). Here we have M = 100, therefore, 100/20 and hence we get our r as 5, which was what initially we started with.

The circuit for implementing Period finding looks like this :



So, after the first QFT on the M inputs, we get the superposition of

$$|0\rangle \xrightarrow{QFT_M} \frac{1}{\sqrt{M}} \sum_{x=0}^{M-1} |x\rangle$$

Which then is acted upon by the function U and hence changing to

$$\frac{1}{\sqrt{M}} \sum_{x=0}^{M-1} |x\rangle |f(x)\rangle$$

So after that according to the principle of deferred measurement, even if we don't measure the qubit before the application of the second QFM circuit, we still get the desired results.

Now Shor's algorithm caters to the problem of finding two odd prime factors of a given number. The difficulty of RSA cryptosystem is based on this. This algorithm involves elementary modular arithmetic.

We find X, such that $X^2 = 1 (\text{mod } N)$, such that X is not equal to 1. Now, we can re-write this equation as $X^2 - 1^2 = 0 (\text{mod } N)$. Hence $(X-1)(X+1) = 0 (\text{mod } N)$. This signifies that, factors of N divides X-1 and X + 1, such that one part of the factors divide one part and the other part

divides the X + 1, where in fact N doesn't divide any of them. So we calculate gcd(N, X+1) and gcd(N, X-1) to get the factors.

Taking an example, say $X^2 = 1(\text{mod } 21)$, Now apart from X = 1, X can be 8 such that 64 = 1(mod 21). Hence we can have 9*7 = 0(mod 21); Now, prime factors of 21, 3 and 7 they divide 8+1 and 8-1 respectively, individually but 21 doesn't divides any of them.

And then we calculate the gcd of (21, 8+1) = 3 and gcd(21, 8-1) = 7, Hence we get the prime factors. So if we have a closer look at it we can see that instead of **$X^2 = 1(\text{mod } N)$(eq 1)**, there's a better method to compute this modulus function to search for **$(X^{r/2})^2 \sim 1(\text{mod } N)$(eq 2).** Consider this by an example that say we have N = 21 and X = 2 from equation 1, then following the pattern of exponents and modulus function, we find this behavior :

$2^0 = 1(\text{mod } 21)$

$2^1 = 2(\text{mod } 21)$

$2^2 = 4(\text{mod } 21)$

$2^3 = 8(\text{mod } 21)$

$2^4 = 16(\text{mod } 21)$

$2^5 = 11(\text{mod } 21)$

$2^6 = 1(\text{mod } 21)$

We see that this is periodic. Randomly picking up a value : $= 2^6 \Rightarrow (2^3)^2 = 1(\text{mod } 21)$ could be a possible solution. Therefore, if we pick, X at random such that $X^r = 1 \text{ (mod N)}$, Then we need to satisfy two conditions, firstly that r is even and hence, **$(X^{r/2})^2 \sim 1(\text{mod } N)$** and secondly that $X^{r/2}$ is not equivalent to **1(mod N).** The probability of getting this on random selection is atleast 0.5, which is quite a compelling case. This is used in the algorithm.

N = 21

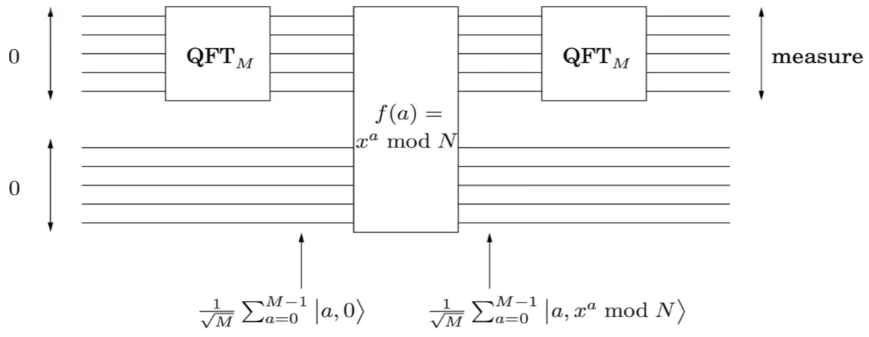
X = 2

a	$f(a) = X^a(\text{mod } N)$
0	1

1	2
2	4
3	8
4	16
5	11
6	1
7	2
8	4
9	8
10	16
11	11
12	1
13	2
14	4

We see that, the r we are looking for in $\mathbf{(X^{r/2})^2 \sim 1 (mod\ N)}$, is nothing but the period.

So now, we just implement the period finding in this algorithm.



Hence, the proof of Shor's Algorithm.

Grover's Algorithm

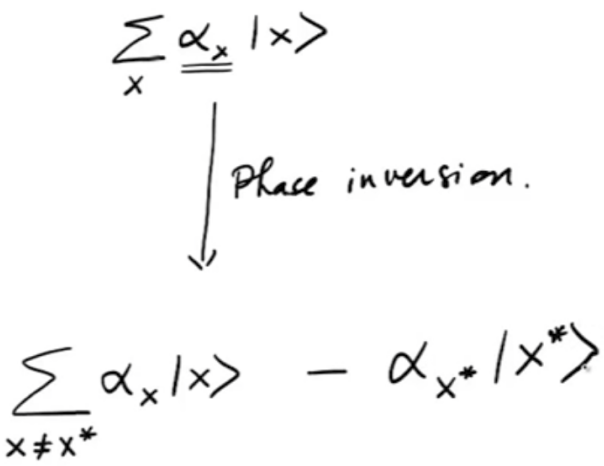
Grover's Algorithm is for unstructured search, just like searching for a needle in a haystack, digital equivalent being a very large table. Randomly picking out entries yield O(N/2) time complexity, where $N = 2^n$. This problem of search can be considered equivalent to the NP-complete problem of Satisfiability, where we have a function defined as f(x1,x2,..,xN) being satisfied or being evaluated to 1 for one of the 2^n inputs, and we need to find that value from the domain.

Grover's algorithm solves this problem of unstructured search in $O(N^{1/2})$. So we have a function f(x) such that f:{0, 1, .., N-1}→{0,1}. Hardest case being f(x) = 1 for just one value in domain.

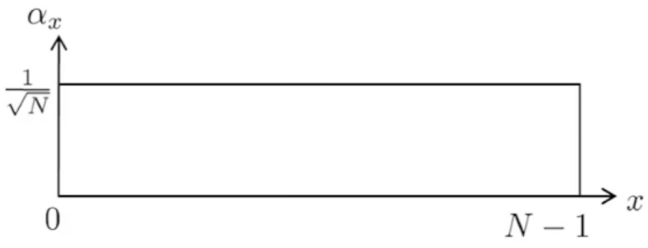
There are two steps to this algorithm :

1. Phase Inversion

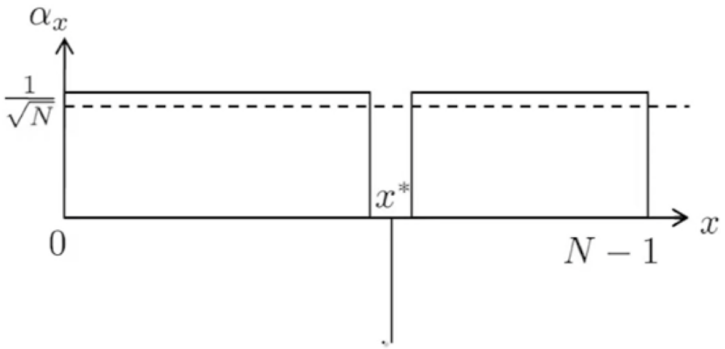
   So at any point of time in this algorithm we have amplitudes in superposition of x. So this step basically performs the phase inversion such that if for x* (f(x*) = 1), is does :



   else, if x is a normal element it leaves it as such. Consider the following graph for f(x):



Then after phase inversion, our function would be something like this.



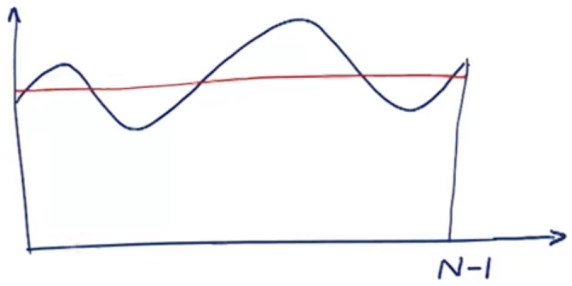
2. Inversion about Mean

Here we flip the amplitudes about the mean, which is calculated as :

$$\mu = \frac{\sum_{x=0}^{N-1} \alpha_x}{N}$$

[2]

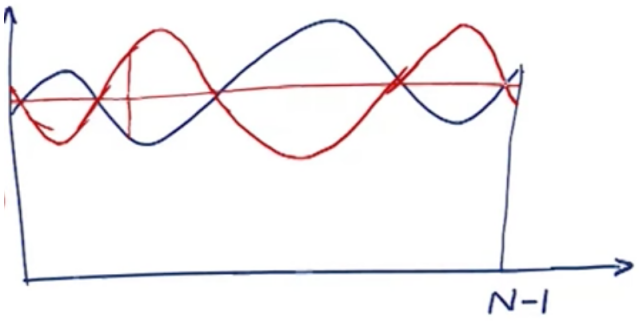
Consider the graph of the function as :



$$\alpha_x \longrightarrow (2\mu - \alpha_x).$$

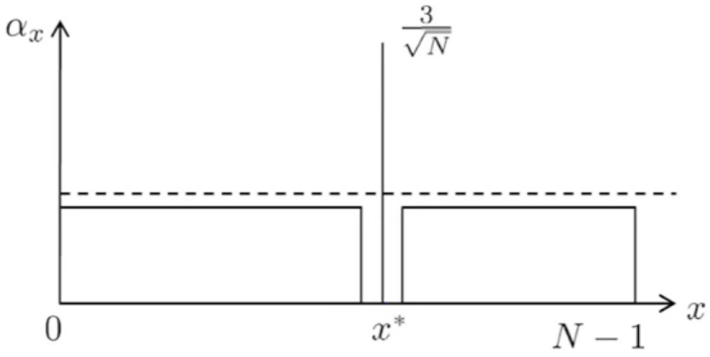
$$\sum_x \alpha_x |x\rangle \longrightarrow \sum_x (2\mu - \alpha_x)|x\rangle$$

Inverted f(x) about the mean looks like this :



Implementation of Grover's Algorithm :

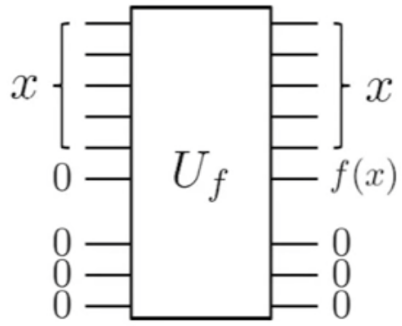
If we do the Mean inversion after the phase inversion, then our function could be represented as



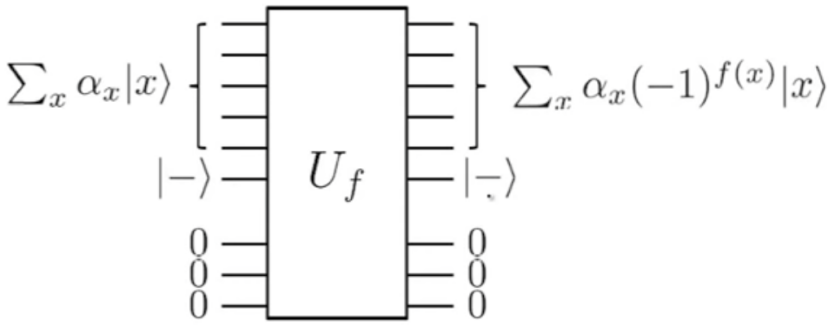
We keep doing these steps, phase inversion followed by mean inversion, we increase the amplitude of x* by $2/N^{1/2}$. Correspondingly, if we keep going this way increasing $5/N^{1/2}$, $7/N^{1/2}$, $9/N^{1/2}$, .. , $1/2^{1/2}$, in $N^{1/2}$ steps. Now the probability of required element location among elements comes to be 0.5 and hence, we can find that. **[1] [5] [2]**

$$\sum_x \alpha_x |x\rangle \xrightarrow[\text{inversion}]{\text{Phase}} \sum_x \alpha_x (-1)^{f(x)} |x\rangle$$

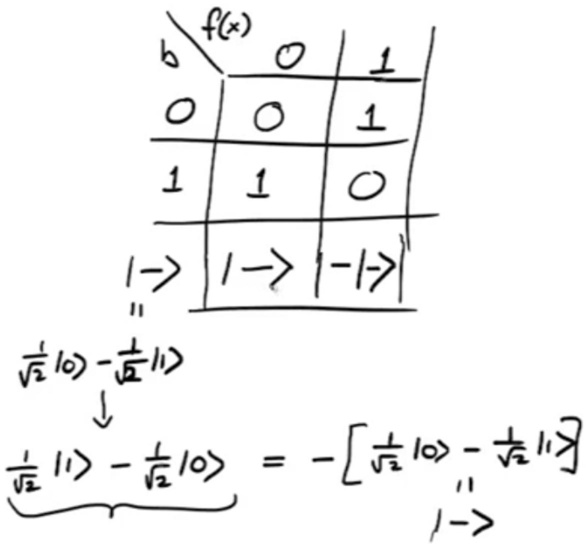
So to get above superposition, in the following circuit,



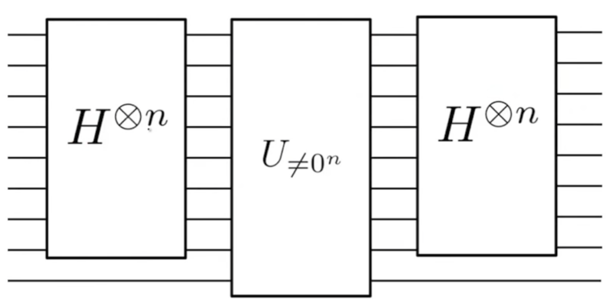
We change the input bit 0 to |->, to perform the inversion. **[2] [1]**



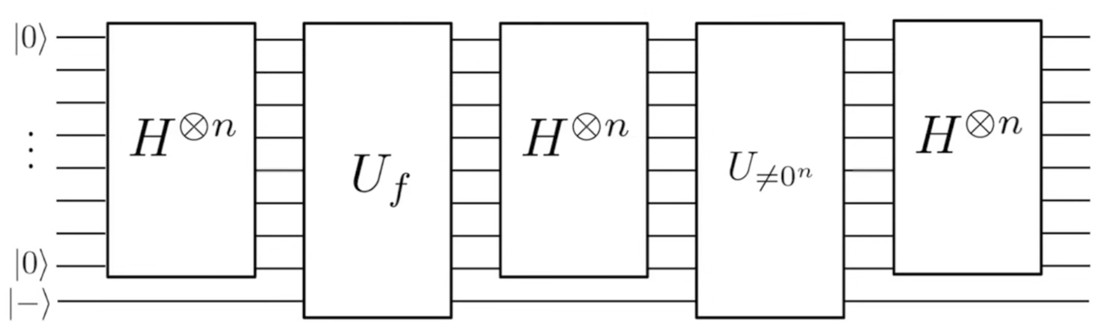
Mathematically, lets consider that bit to be say b, then output is b XOR f(x).



and we can associate this calculated negative sign in the superposition as one associated with inversion of phase when x = x*. Circuit for calculating inversion about mean **[2]**:



Hence the final circuit : **[5]**



Deutsch-Jozsa Algorithm

One of the first examples of a quantum algorithm that is exponentially faster than any possible deterministic classical algorithm. It is also a deterministic algorithm, meaning that it always produces an answer, and that answer is always correct.

In the Deutsch-Jozsa problem, we are given a black box quantum computer known as an oracle that implements some function . In layman's terms, it takes n-digit binary values as input and produces either a 0 or a 1 as output for each such value. We are promised that the function is either constant (0 on all inputs or 1 on all inputs) or balanced[3] (returns 1 for half of the input domain and 0 for the other half); the task then is to determine if is constant or balanced by using the oracle.

Works Cited

[1] N. a. Chuang, Quantum Computation and Quantum Information.

[2] U. Vazirani, "https://people.eecs.berkeley.edu/~vazirani/f16quantum.html," [Online]. Available: https://people.eecs.berkeley.edu/~vazirani/f16quantum.html.

[3] S. Bhambri, "Quantum Clouds : A future perspective," *Arxiv.org,* 2014.

[4] Z. Goodwin. [Online]. Available: http://www.physlink.com/education/askexperts/ae329.cfm.

[5] J. D. Jackson, "Mathematics for Quantum Mechanics: An Introductory Survey of Operators, Eigenvalues, and Linear Vector Spaces".

[6] "Wiki," [Online]. Available: https://en.wikipedia.org/wiki/Expectation_value_(quantum_mechanics).

[7] jcresser. [Online]. Available: http://physics.mq.edu.au/~jcresser/Phys301/Chapters/Chapter13.pdf.