Quantum States..

- A bit is 0 or 1, a qubit is in a superposition of|0> and|1>

$$\psi\rangle = \alpha_0 \, |0\rangle + \alpha_1 \, |1\rangle$$

- If we measure then we get one outcome.

The probability of measuring |0> is $|\propto 0|^2$

The probability of measuring |1> is $|\propto 1|^2$

- We combine qubits to create bigger states via Tensor Products.
- Quantum states are normalized complex vectors.

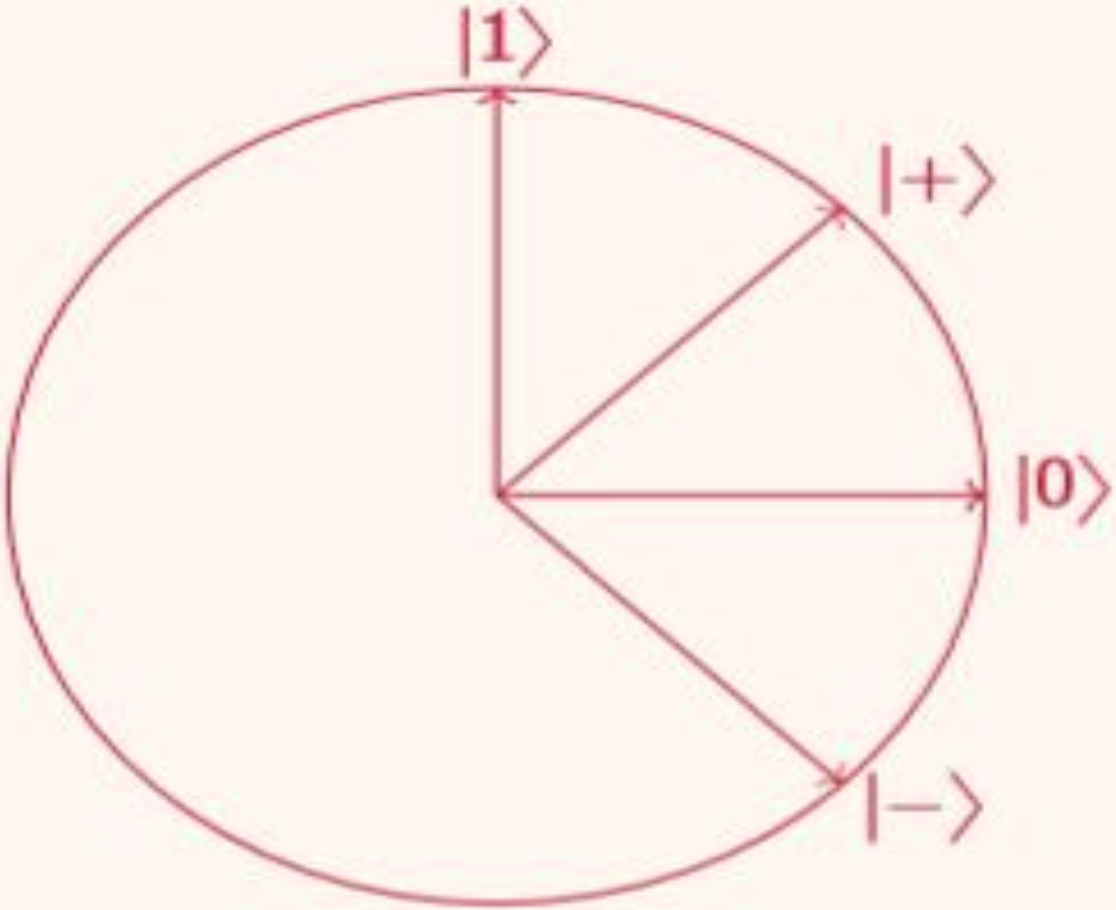
Quantum Gates…

- We can change states by applying unitaries, they keep vector normalized.
- Unitaries on a only a few qubits are called gates.
- I does nothing .
- X changes |0> into |1> and vice versa.
- Z adds a -1 in front of |1>, and
- Z is just X in the {|+>,|->} basis (and vice versa).
- H changes |0> and |1> into $\frac{|0>+|1>}{\sqrt{2}}$ and $\frac{|0>-|1>}{\sqrt{2}}$.
- H is a basis transform.

$$I := \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad X := \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix},$$

$$Z := \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \quad H := \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}.$$

$$|+\rangle := \frac{|0\rangle + |1\rangle}{\sqrt{2}} \qquad |-\rangle := \frac{|0\rangle - |1\rangle}{\sqrt{2}}$$



Phase Shift Gate...

- A phase is a factor of the form $e^{i2\pi\theta}$.
- Phase Shift Gate : It is a single qubit basis gate it maps |0> to |0> and |1> to $e^{i2\pi\theta}$|1>.
- The probability of measuring |0> or |1> is unchanged after applying the phase ,however it changes the phase of the quantum state .

$$P(\varphi) = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\varphi} \end{bmatrix}$$

$$Z = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} = P(\pi)$$

$$S = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\frac{\pi}{2}} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix} = P\left(\frac{\pi}{2}\right) = \sqrt{Z}$$

$$T = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\frac{\pi}{4}} \end{bmatrix} = P\left(\frac{\pi}{4}\right) = \sqrt{S} = \sqrt[4]{Z}$$

The Controlled Gates…

- Controlled gates act on 2 or more qubits, where one or more qubits act as a control for some operation. For example, the controlled NOT gate (or CNOT or CX) acts on 2 qubits, and performs the NOT operation on the second qubit only when the first qubit is |1> .

Controlled Phase Shift Gate…

Controlled phase shift [edit]

The 2-qubit controlled phase shift gate is:

$$\mathrm{CPHASE}(\varphi) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & e^{i\varphi} \end{bmatrix}$$

With respect to the computational basis, it shifts the phase with φ only if it acts on the state $|11\rangle$:

$$|a,b\rangle \mapsto \begin{cases} e^{i\varphi}|a,b\rangle & \text{for } a=b=1 \\ |a,b\rangle & \text{otherwise.} \end{cases}$$

The CZ gate is the special case where $\varphi = \pi$.

Oracles…

- Classical algorithm make calls to the memory to get the input.
- Quantum algorithms get an oracle that mimics this.
- Oracle is also called as “Black Box”.
- A binary oracle for an input $x \in \{0,1\}$^n is a unitary

$$O_x|i\rangle|b\rangle = |i\rangle|b \oplus x_i\rangle$$

- A phase oracle for an input $x \in \{0,1\}$^n is a unitary

$$O_{x,\pm}|i\rangle = (-1)^{x_i}|i\rangle$$

- A Controlled phase oracle:

$$O_{x,\pm}|i\rangle|0\rangle = |i\rangle|0\rangle, \quad O_{x,\pm}|i\rangle|1\rangle = (-1)^{x_i}|i\rangle|1\rangle$$

Quantum Algorithms

The Deutsch's Algorithm

- It is an algorithm designed for the execution on quantum computers and has a potential to be more efficient than classical algorithm by taking the advantage of Quantum Superposition and Entaglement.
- Deutsch's algorithm determines if the given function is constant or balanced.
- Constant Function: f(0)=f(1).
- Balanced Function : f(0)!=f(1).

- Classically we need to evaluate both f(0) and f(1).

The problem Setting

- "f" is a function defined over the binary values 0,1. such that

  f: 0,1 $\longrightarrow$ 0,1.

- There are four possible configurations for the function f:
- Both inputs 0,1 are mapped to the output 0: f(0)=f(1)=0.
- Both inputs 0,1 are mapped to the output 1: f(0)=f(1)=1.
- Inputs pass through f unchanged: f(0)=0 and f(1)=1.
- Inputs are exchanged after passing through f: f(0)=1 and f(1)=0.

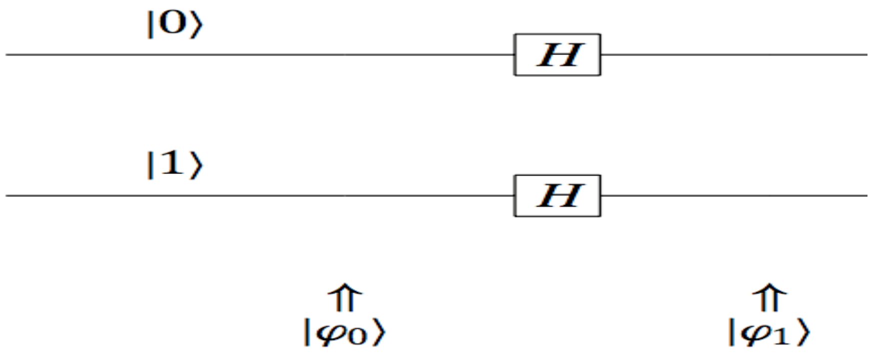
The first cases as "f" being "constant",and last two cases as f being "balanced".

The problem: Given a function **f:0,1⟶0,1** and without knowing anything more than that determine whether "f" is a constant or a balanced function with the minimum number of function evaluation.

- In classical way we have to evaluate each case.

Procedure:

- Step 1 : Apply Hadamard Gate to the input state $|0>|1>$, to produce a product state of two superpositions.



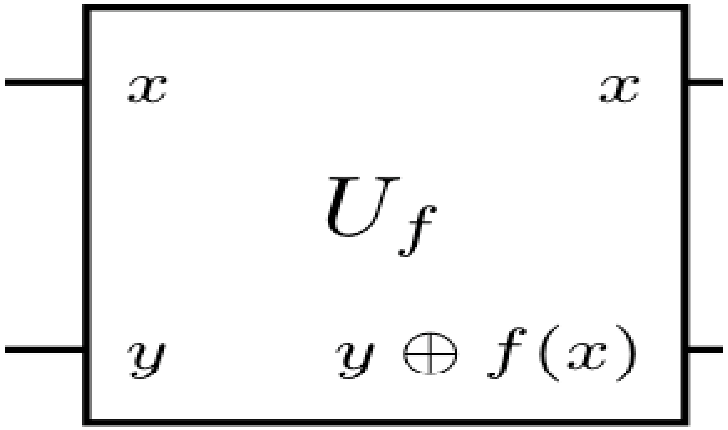
$$|\varphi 0> = |0>|1>.$$

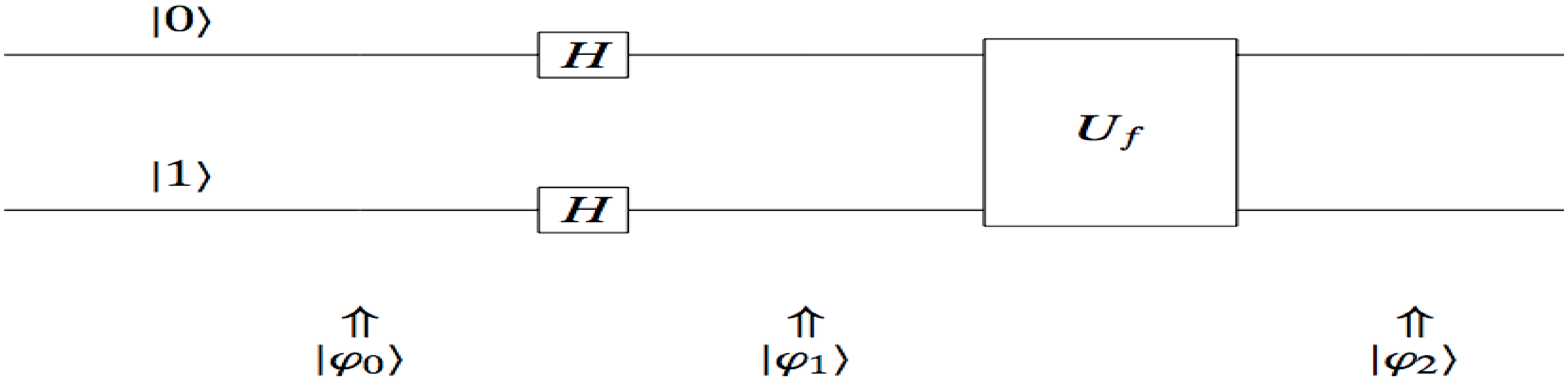
$$|\varphi 1> = \left(\frac{|0>+|1>}{\sqrt{2}}\right)\left(\frac{|0> -|1>}{\sqrt{2}}\right) = \frac{1}{2}(|00> - |01> + |10> - |11>)$$

first qubit is in a superposition representing both possible input $|0\rangle$ and $|1\rangle$ for the Oracle function.

- Step 2: Apply “Uf” to the product state($|\varphi 1>$).
- “Uf” is an unitary operation that acts on two qubits (also called “oracle”). It leaves the first qubit alone and produce the “exclusive or(XOR)” of the second qubit with the function “f” evaluated with the first qubit as argument.

$$\boldsymbol{Uf|x,y> = |x,y\ XOR\ f(x)>}$$





The output become = $$|\varphi_2\rangle = \frac{(-1)^{f(0)}|0\rangle + (-1)^{f(1)}|1\rangle}{\sqrt{2}} \otimes \frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

$$(-1)^{f(0)}|0\rangle + (-1)^{f(1)}|1\rangle$$

$$(-1)^{f(x)}|x\rangle(|0\rangle - |1\rangle)$$

$$|0\rangle + |1\rangle$$

$$(-1)^{f(0)}|0\rangle + (-1)^{f(1)}|1\rangle$$

If "f" is constant the above expression either become

$$+1(|0> +|1>)\ or\ -1(|0> +|1>)$$

If "f" is balanced the above expression either become

$$+1(|0> -|1>)\ or\ -1(|0> -|1>)$$

- So the output will become=

$$|\varphi_2\rangle = \begin{cases} (\pm 1)\frac{|0\rangle+|1\rangle}{\sqrt{2}} \otimes \frac{|0\rangle-|1\rangle}{\sqrt{2}}, & \text{if } f \text{ is constant,} \\ (\pm 1)\frac{|0\rangle-|1\rangle}{\sqrt{2}} \otimes \frac{|0\rangle-|1\rangle}{\sqrt{2}}, & \text{if } f \text{ is balanced.} \end{cases}$$

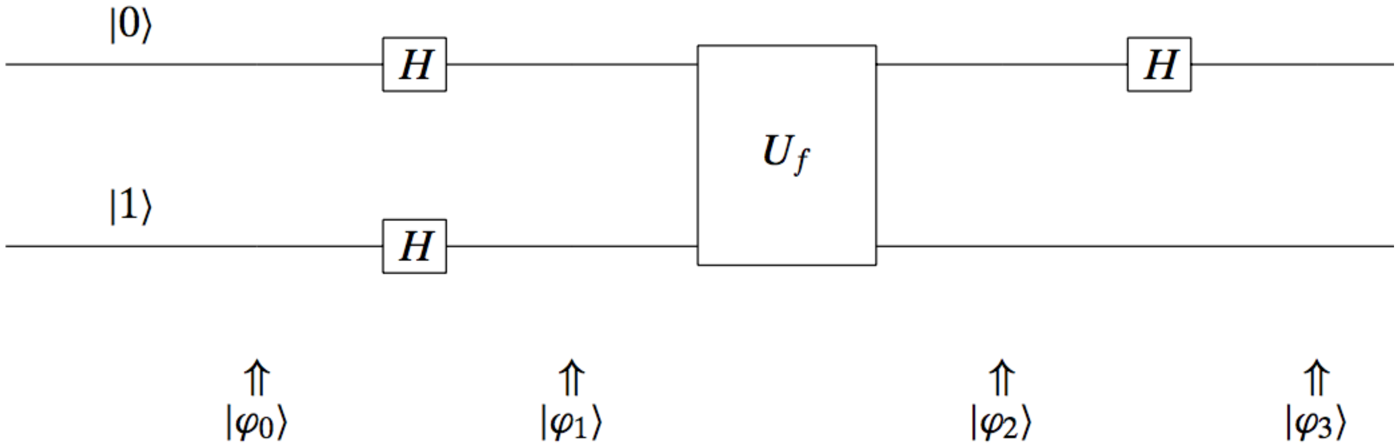
- Constant Oracle:When the oracle is *constant*, it has no effect (up to a global phase) on the input qubits, and the quantum states before and after querying the oracle are the same.

- Balanced Oracle:When the oracle is *balanced*, phase kickback adds a negative phase to exactly half these states.

- Step 3(Final step): Apply a Hadamard Gate to the first qubit leaving the second qubit alone.

$$H(H|0>) = H|+>= |0>$$

$$H(H|1>) = H|->= |1>$$



$$|\varphi_3\rangle = \begin{cases} (\pm 1)|0\rangle \otimes \frac{|0\rangle - |1\rangle}{\sqrt{2}}, & \text{if } f \text{ is constant,} \\ (\pm 1)|1\rangle \otimes \frac{|0\rangle - |1\rangle}{\sqrt{2}}, & \text{if } f \text{ is balanced.} \end{cases}$$

If the output is 0 then it is a "constant function".

If the output is 1 then, it is a " balanced function"

Example:

- Using two qubit.

Consider a two-bit function $f(x_0, x_1) = x_0 \oplus x_1$ such that

$f(0,0) = 0$

$f(0,1) = 1$

$f(1,0) = 1$

$f(1,1) = 0$

The corresponding phase oracle of this two-bit oralce is $U_f|x_1, x_0\rangle = (-1)^{f(x_1,x_0)}|x\rangle$

- Step 1 : $|\varphi 0> = |00> |1>$

- Step 2: Applying hadamard on all qubit.

$$|\varphi 1> = \frac{1}{2}(|00> +|01> +(10> +|11>) \otimes \frac{|0> -|1>}{\sqrt{2}}$$

$$|\varphi 2> = \frac{(-1)^{f(0,0)}|00>+(-1)^{f(0,1)}|01>+(-1)^{f(1,0)}|10>+(-1)^{f(1,1)}|11>}{2} \otimes \frac{|0>-|1>}{\sqrt{2}}$$

$$|\varphi 2> = (|00> -|01> -|10> +|11> \otimes \frac{|0> -|1>}{\sqrt{2}}$$

$$= \frac{1}{\sqrt{2}}(|0>-|1>) \otimes \frac{1}{\sqrt{2}}(|0>-|1>) \otimes \frac{|0>-|1>}{\sqrt{2}}$$

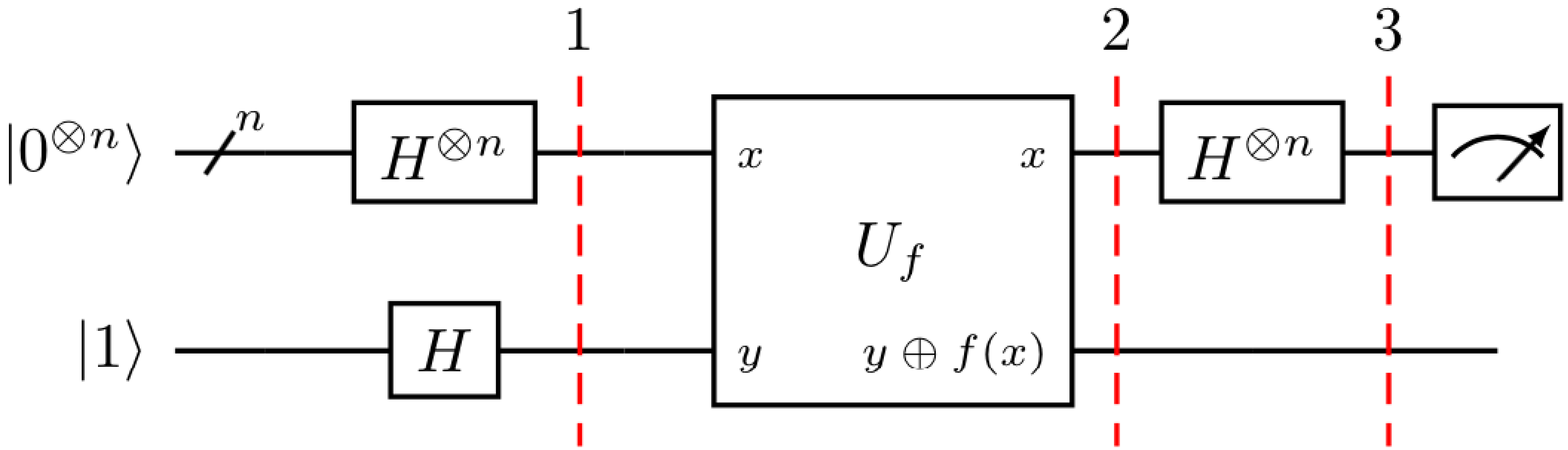
- Step 3 : Applying Hadamard gate on the first qubit register.

$$|\psi_3\rangle = |1\rangle_0 \otimes |1\rangle_1 \otimes (|0\rangle - |1\rangle)_2$$

- Measuring the first two qubits will give the non-zero 11, indicating a balanced function.

Extension to multivariate functions: The Deutsch-Jozsa Algorithm

- The Deutsch-Jozsa algorithm is a generalization of Deutsch's Algorithm.
- This algorithm allows us to determine whether a function f(x) is constant or balanced.
- Here constant f defined as the case where all the outputs are mapped into either 0 or 1 .Balanced f is referred to the case where half of the output 0 ,and the other half goes to 1.


1
2
3
$|0^{\otimes n}\rangle$
n
$H^{\otimes n}$
$|1\rangle$
H
x
x
U_f
y
$y \oplus f(x)$
$H^{\otimes n}$

- Applying *n* Hadamard gates to prepare the superposition, And apply the Oracle function.

$|x\rangle|y\rangle$ to $|x\rangle|y \oplus f(x)\rangle$ apply oracle function

$$\frac{1}{\sqrt{2^{n+1}}}\sum_{x=0}^{2^n-1}|x\rangle(|0\rangle-|1\rangle) \longrightarrow \frac{1}{\sqrt{2^{n+1}}}\sum_{x=0}^{2^n-1}|x\rangle(\underbrace{|0\oplus f(x)\rangle-|1\oplus f(x)\rangle}_{y\oplus f(x)})$$

$$\frac{1}{\sqrt{2^{n+1}}}\sum_{x=0}^{2^n-1}|x\rangle(|f(x)\rangle-|1\oplus f(x)\rangle)$$

equivalent

$$\frac{1}{\sqrt{2^{n+1}}}\sum_{x=0}^{2^n-1}(-1)^{f(x)}|x\rangle(\underbrace{|0\rangle-|1\rangle}_{\text{second qubit}})$$

- The second qubit above will remain the same for the rest of the circuit.

- Applying *n* Hadamard gate again.

The general equation for the Hadamard gate transformation on multiple qubits is:

$$\xrightarrow{H^{\otimes n}} \frac{1}{\sqrt{2^n}} \sum_{z \in \{0,1\}^n} (-1)^{\langle x,z \rangle} |z\rangle$$

- After applying the hadamard gate ,the qubits become

Apply Hadamard gates $\longrightarrow$ $$\frac{1}{2^n}\sum_{x=0}^{2^n-1}(-1)^{f(x)}\left[\sum_{y=0}^{2^n-1}(-1)^{x\cdot y}|y\rangle\right]$$

$$\frac{1}{2^n}\sum_{y=0}^{2^n-1}\left[\sum_{x=0}^{2^n-1}(-1)^{f(x)}(-1)^{x\cdot y}\right]|y\rangle$$

where $x \cdot y = x_0 y_0 \oplus x_1 y_1 \oplus \cdots \oplus x_{n-1} y_{n-1}$ is the sum of the bitwise product.

- If measurement output is all zeroes then f is constant, for any other output f is balanced.

Bernstein-Vazirani Algorithm

- It is an extension of the Deutsch-Jozsa algorithm.
- If we have a oracle and n-bit string is hidden in it , To find that string classical computation takes “n” times , But in Quantum computers it will only take 1 query.
- **The Bernstein-Vazirani Problem:**Bernstein Vaziran’s algorithm is related to finding the black box function called Oracle. It has a string of bits which is based on a secret string. The goal of the algorithm is to find the string of bits which gives us the dot product of the oracle string.
- The classical algorithm for finding the secret string is to use m bits one to find out each bit in the secret string.

Procedure:

- Assume our secret string is s with n bits.
- Step 1: Initialize the input qubit as $|0>^{\otimes n}$ and |->.
- Step 2: Applying hadamard gate to the first n qubits.

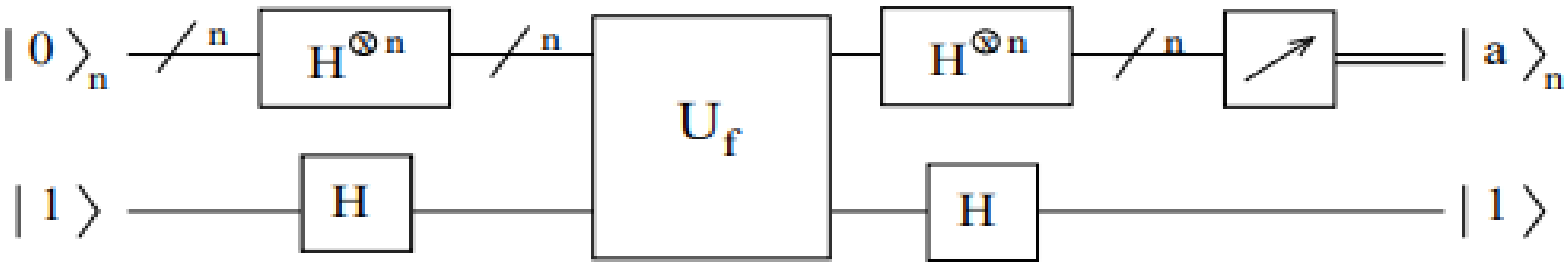
$$|00\ldots0\rangle \xrightarrow{H^{\otimes n}} \frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} |x\rangle$$

- Step 3: Applying a oracle f_a having the secret number "a".

$$|00\ldots0\rangle \xrightarrow{H^{\otimes n}} \frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} |x\rangle \xrightarrow{f_a} \frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} (-1)^{a\cdot x}|x\rangle$$

- Step 4: Applying Hadamard gate again.

$$\frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} (-1)^{a \cdot x} |x\rangle \xrightarrow{H^{\otimes n}} |a\rangle$$



Example: 2 Qubit

- Step 1: $|\varphi 0> = |00>$
- Step 2: Applying Hadamard gate,

$$|\varphi 1> = H|00> = \frac{1}{2}(|00> + |01> + |10> + |11>)$$

- Step 3: Secret string s=11,Applying Quantum Oracle function.

$$|\varphi 2> = \frac{1}{2}\Big((|0> + -1^{s1}|1>) \otimes (|0> + -1^{s2}|1>)\Big)$$

$$= \frac{1}{2}\Big((|0 \succ -|1>) \otimes (|0 \succ -|1>)\Big)$$

$$= \frac{1}{2}(|00 \succ -|01 \succ -|10> + |11>)$$

- Step 4:Applying Hadamard Gate to the first register.

$$\mathrm{H}(\frac{1}{2}(|00 \succ -|01 \succ -|10> + |11>)) = |11>$$

$$H^{\otimes 2}|00\rangle = \tfrac{1}{2}(|00\rangle + |01\rangle + |10\rangle + |11\rangle)$$

$$H^{\otimes 2}|01\rangle = \tfrac{1}{2}(|00\rangle - |01\rangle + |10\rangle - |11\rangle)$$

$$H^{\otimes 2}|10\rangle = \tfrac{1}{2}(|00\rangle + |01\rangle - |10\rangle - |11\rangle)$$

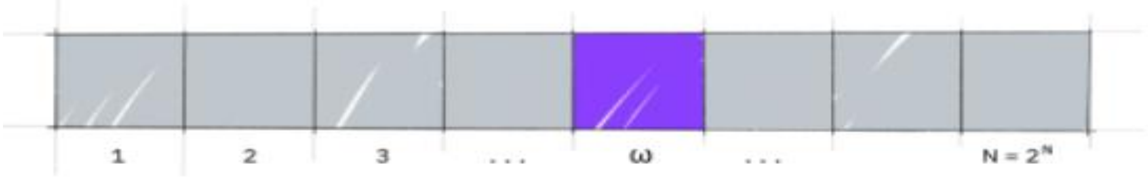
$$H^{\otimes 2}|11\rangle = \tfrac{1}{2}(|00\rangle - |01\rangle - |10\rangle + |11\rangle)$$

The Grover's Algorithm

- it can be used to solve unstructured search problems.
- This algorithm can speed up an unstructured search problem quadratically.
- For an unstructured search ,In classical computation it may take an average on **N /2.** Or may be the worst case will be **N.** But in quantum computation we can find it roughly $\sqrt{N}$ **steps,** using **Grover's amplitude amplification tricks.**
- Taking advantage of qubit superposition and phase interference to improve unstructured database search from O(N) to O($\sqrt{N}$).

Unstructured Search

- Suppose we have a large list of N items. Among these items there is one item with a unique property that we wish to locate that is the winner *"w"*.



- To find the winner *w* using classical computation, one would have to check on average N/2 of these boxes, and in the worst case, all N of them. On a quantum computer, however, we can find the marked item in roughly √N steps with Grover's amplitude amplification trick.

- Step 1: Applying hadamard gate to all the Qubit.

$$H^n|0> = \frac{1}{\sqrt{\sqrt{2}^n}} \sum_{k\{0,1\}} |x> = |s>.$$

|s> is the superposition state.with all the single state having $\frac{1}{\sqrt{\sqrt{2}^n}}$ probability.

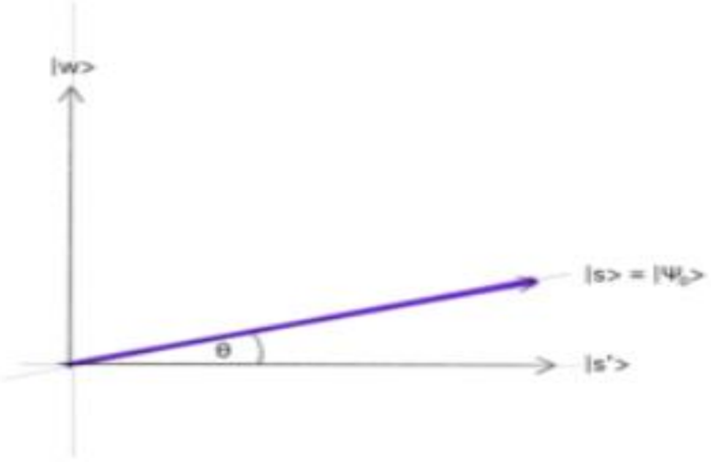
|w> is the winner state,and |s> and |w> are not orthogonal.

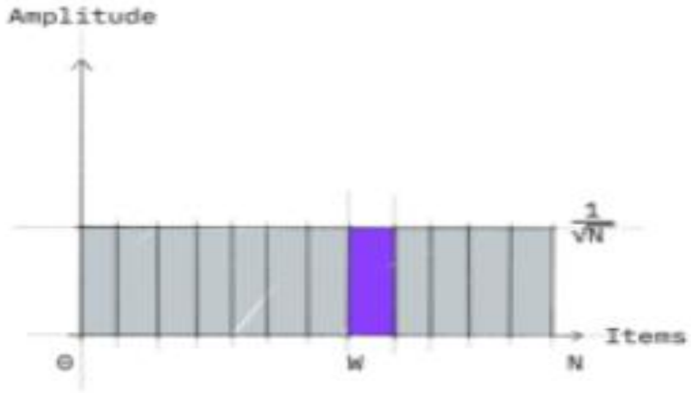
|s’> is the state without the winner.and it’s orthogonal to |w>

Procedure

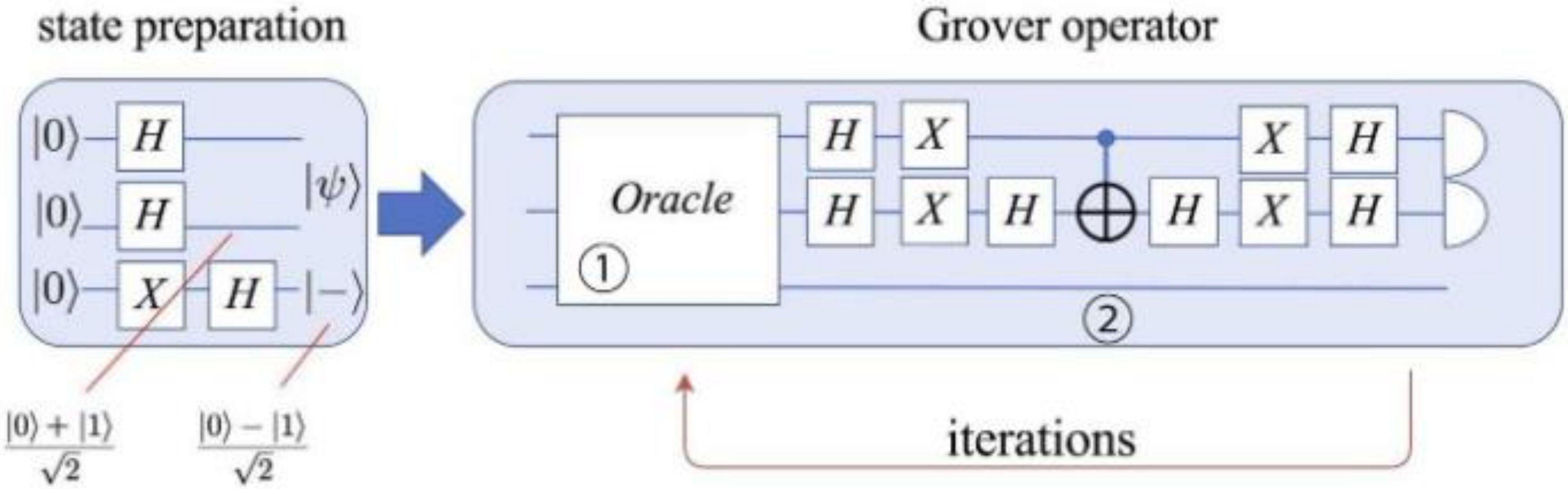
- Step 1: Preparing a superposition state.(The amplitude amplification procedure starts out in the uniform superposition |s>=|ψ>)
-

$$|\psi\rangle = H^{\otimes n} |0\rangle^{\otimes n} = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle$$





Quantum Circuit for Grover's Algorithm.



Amplitude Amplification

- In step 2: we perform amplitude amplification by applying another oracle called reflection operator also called Grover's diffusion operator.
- Amplitude Amplification: Grover's algorithm, implements *amplitude amplification* to increase the probability of observing the correct answer(the object of the search).(**Increases** the probability amplitude associated with answer. **Decreases** all other probability ampliltude)
- Reflection operator : By using reflection operator we can perform amplitude amplification, we can amplify the amplitude of the winning states (w) and reduce the amplitude of the non-winning states.

Creating Oracle…

- Step 2: After the creation of superposition state , the Grover's algorithm turns into an iterative process which composes of multiple iterations of Oracle function and the Grover operator.
- We apply the oracle reflection "Uf" to the state |s⟩.

$$|x\rangle \otimes |q\rangle \xrightarrow{O_f} |x\rangle \otimes |q \oplus f(x)\rangle$$

$$f(x) = \begin{cases} 0 & \text{if } x \neq u \\ 1 & \text{if } x = u \end{cases}$$

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

$$O|\mathbf{x}\rangle \frac{|0\rangle - |1\rangle}{\sqrt{2}} \to |\mathbf{x}\rangle \frac{|f(\mathbf{x}) \oplus 0\rangle - |f(\mathbf{x}) \oplus 1\rangle}{\sqrt{2}}$$

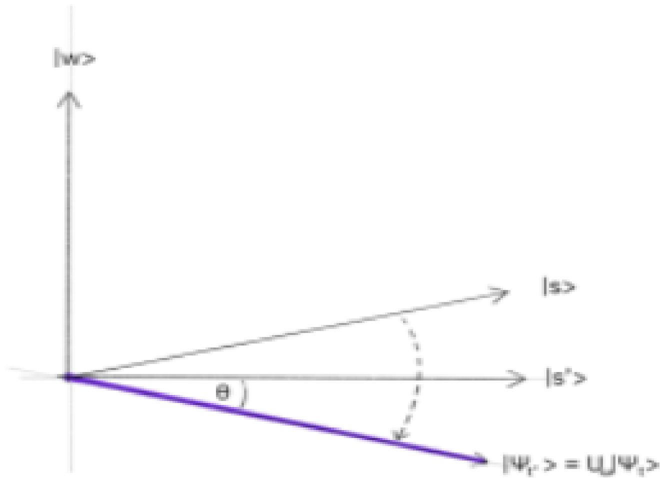
reverse the amplitude if f(x)=1

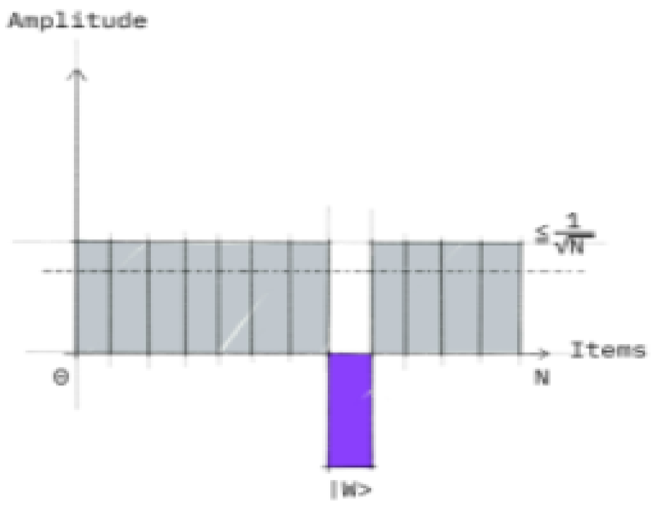
$$\text{if } f(x)=1 \to |\mathbf{x}\rangle \frac{|1 \oplus 0\rangle - |1 \oplus 1\rangle}{\sqrt{2}} = -|\mathbf{x}\rangle \frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

$$\text{if } f(x)=0 \to |\mathbf{x}\rangle \frac{|0 \oplus 0\rangle - |0 \oplus 1\rangle}{\sqrt{2}} = |\mathbf{x}\rangle \frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

no change

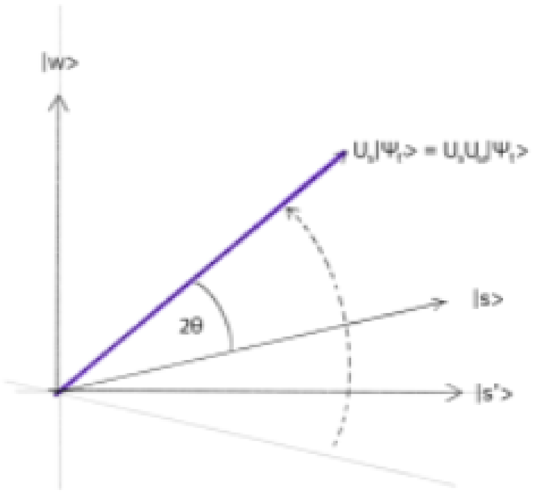
• For any |x>, f(x)=)0 it's does not change. Otherwise the amplitude is changed to negative.

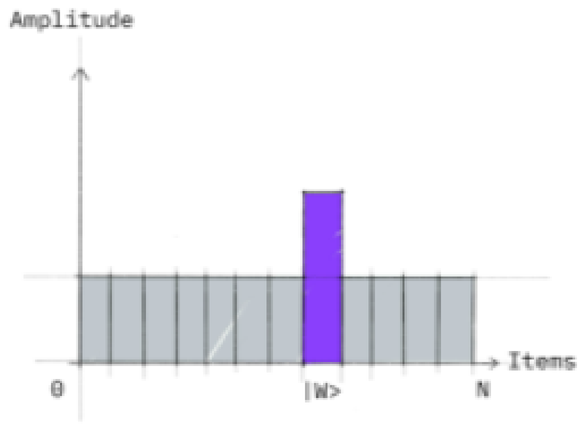




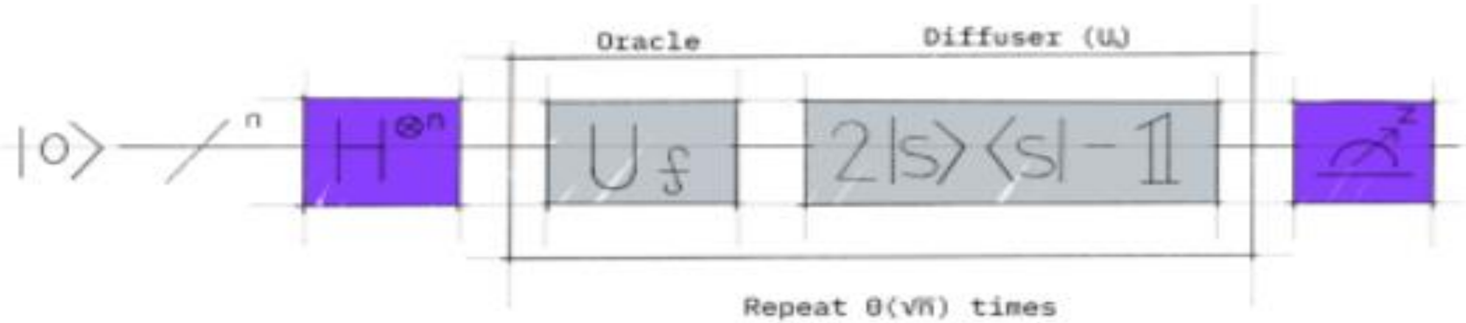
• this corresponds to a reflection of the state |s⟩ about |s'⟩. This transformation means that the amplitude in front of the |w⟩ state becomes negative, which in turn means that the average amplitude (indicated by a dashed line) has been lowered.

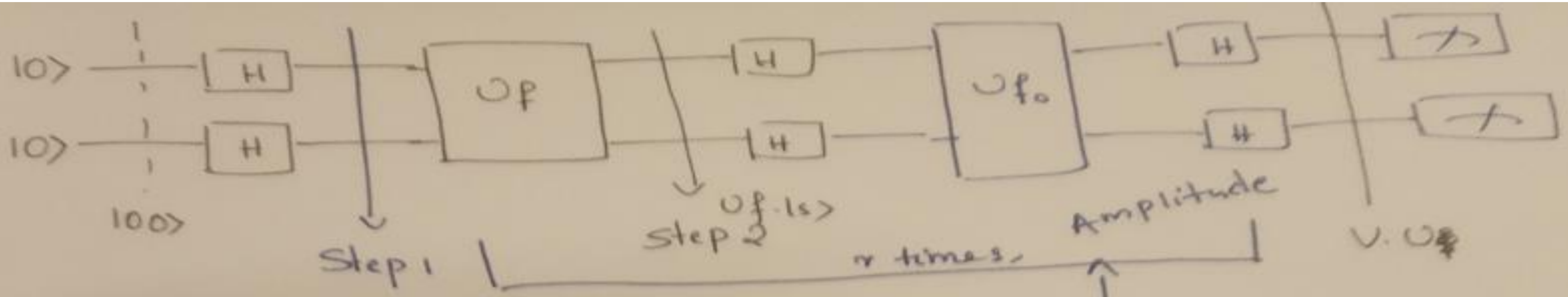
- Step 3 :now applying an additional reflection (Us) about the state |s⟩: Us=2|s⟩⟨s|−I. This transformation maps the state to UsUf|s> and completes the transformation.

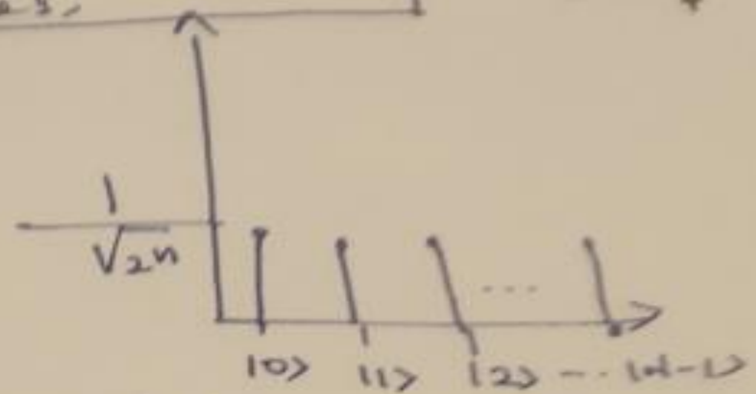




- Two reflections always correspond to a rotation. The transformation ”UsUf” rotates the initial state |s>closer towards the winner |w⟩.
- This procedure will be repeated several times to zero in on the winner.
- After t steps we will be in the state $|\psi t>$ where: $|\psi t\rangle=(UsUf)^\wedge t|s\rangle$.





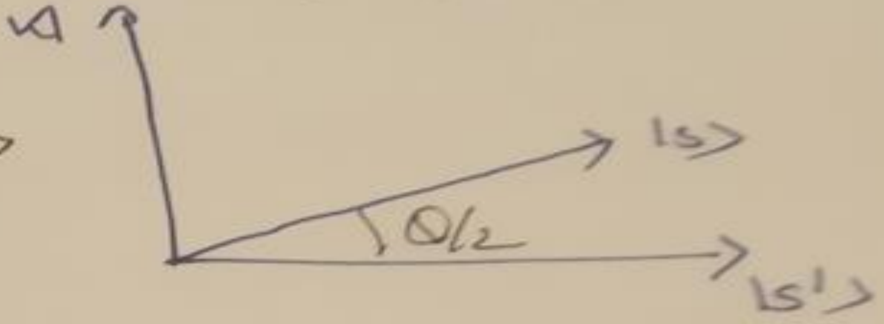


Step 1: Applying Hadamard Gate.

$$H^{\otimes n}|0\rangle = \frac{1}{\sqrt{2^n}}\sum_{x\in\{0,1\}^n}|x\rangle = |s\rangle$$

$$|s'\rangle = \frac{1}{\sqrt{2^n-1}}\sum_{x\neq w}|x\rangle$$

$\cancel{|s\rangle = \cos\theta/2\,|s'\rangle + \sin\theta/2\,|w\rangle}$



Step 2: Phase inversion.

$$U_f|w\rangle = -|w\rangle$$

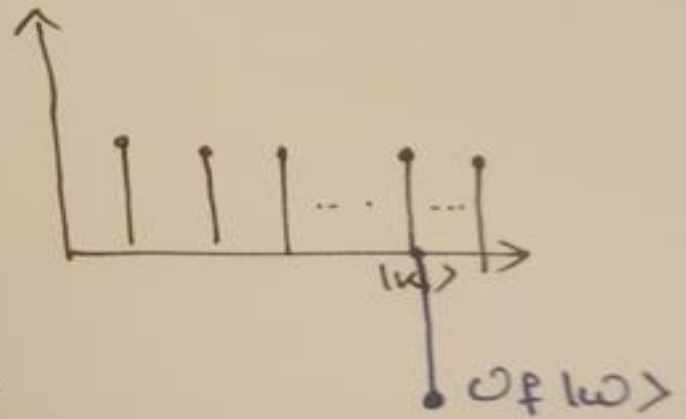
$$U_f|x\rangle = |x\rangle,\ x\neq w$$

$$U_f|s\rangle = (I - 2|w\rangle\langle w|)|s\rangle$$

Step 2: Phase inversion.

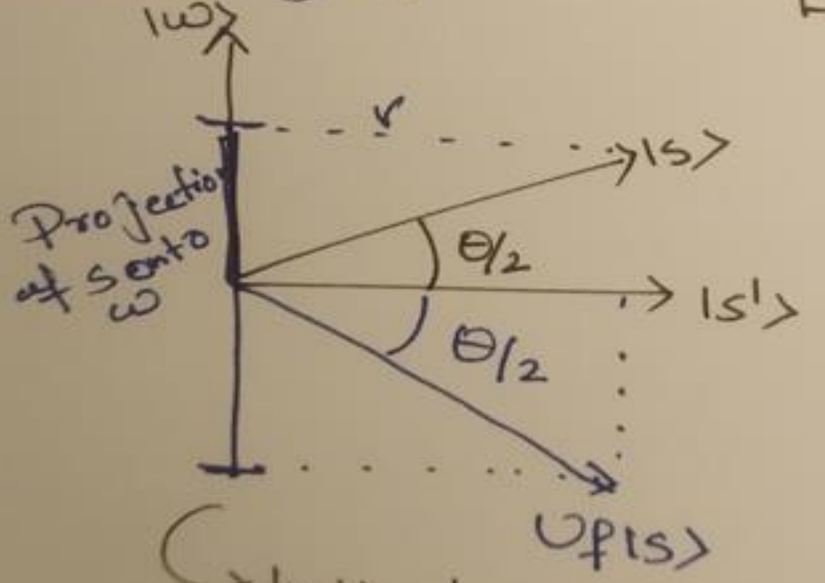
$$U_f|w\rangle = -|w\rangle$$

$$U_f|x\rangle = |x\rangle, \; x \neq w$$



$$U_f|s\rangle = (I - 2|w\rangle\langle w|)|s\rangle$$

$$=$$

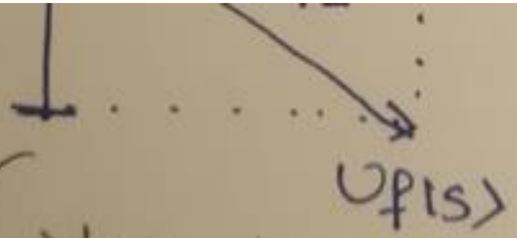


↪ $|w\rangle$ becomes -ve.

Step 3: Inversion about Mean.

$V = 2|s\rangle\langle s| - I$ (Projection on $U_f|s\rangle$ onto $|s\rangle$)

→ two times → $VU_f|s\rangle$



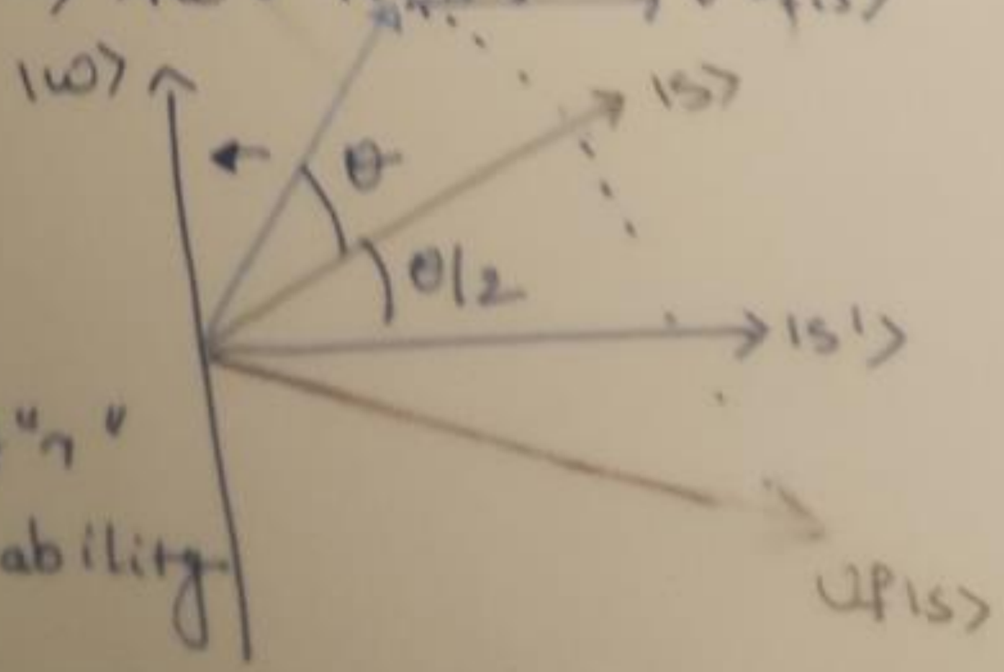
$\hookrightarrow |k1\rangle$ becomes -ve.

Step 3: Inversion about Mean.

$V = 2|s\rangle\langle s| - I$ (Projection on $U_f|s\rangle$ onto $|s\rangle$)

→ two times → $VU_f|s\rangle$

$VU_f|s\rangle =$



It shifted $\theta + \frac{\theta}{2} = 3\theta/2$

$(V.U_f)^r|s\rangle$ will repeated about "r" times to get high probability ~~that~~, $|w\rangle$

Example : 2 Qubit

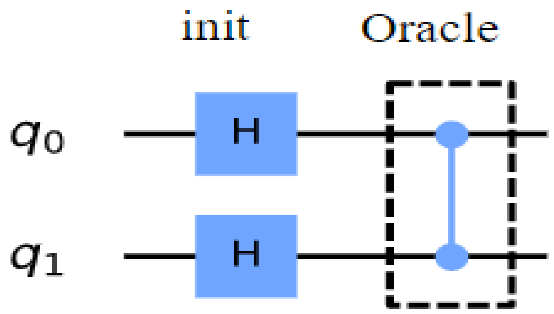
- The winner sate is |w>=3=|11>
- Oracle for |w>

$$U_\omega|s\rangle = U_\omega\frac{1}{2}(|00\rangle + |01\rangle + |10\rangle + |11\rangle) = \frac{1}{2}(|00\rangle + |01\rangle + |10\rangle - |11\rangle)\,.$$

or:

$$U_\omega = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}$$

oracle is the controlled Z gate.



- Reflection:In order to complete the circuit we need to implement the additional reflection Us=2|s⟩⟨s|−1. Since this is a reflection about |s⟩, we want to add a negative phase to every state orthogonal to |s⟩.
- One way we can do this is to use the operation that transforms the state |s⟩→|0⟩.

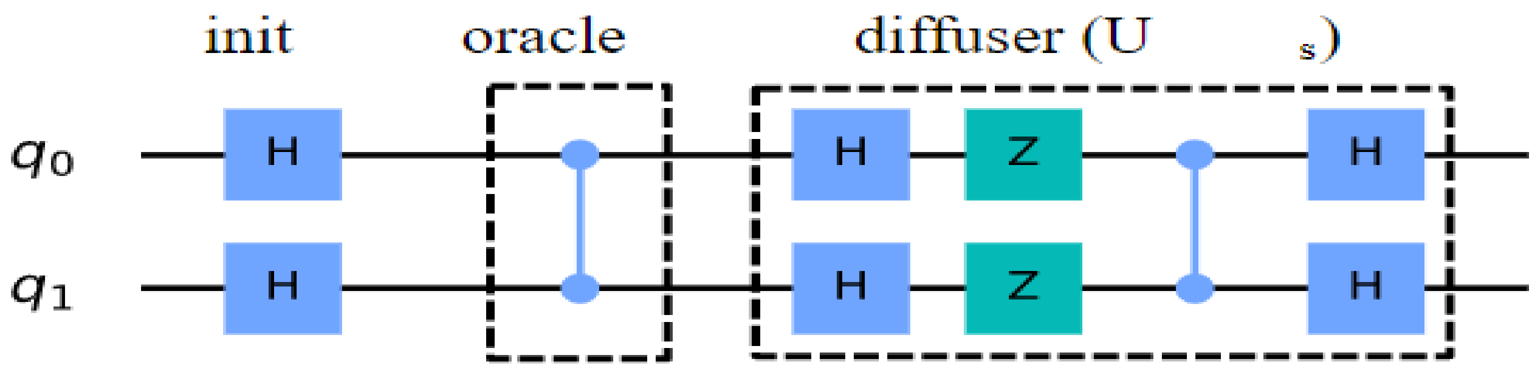
$$H^{\otimes n}|s\rangle = |0\rangle$$

- Then we apply a circuit that adds a negative phase to the states orthogonal to |0⟩.

$$U_0 \frac{1}{2}(|00\rangle + |01\rangle + |10\rangle + |11\rangle) = \frac{1}{2}(|00\rangle - |01\rangle - |10\rangle - |11\rangle)$$

- Finally, we do the operation that transforms the state |0⟩→|s⟩ (the H-gate again):

$$H^{\otimes n} U_0 H^{\otimes n} = U_s$$



Shor's Algorithm

- Shor’s algorithm is famous for factoring integers in polynomial time
- There are two components in the Shor's Algorithm .

1. Quantum Fourier Transform
2. Quantum Phase Estimation(that used Quantum Fourier Transformation).

Quantum Fourier Transform

- The quantum Fourier transform (QFT) is the quantum implementation of the discrete Fourier transform over the amplitudes of a wavefunction.
- Fourier transform is the mathematical tool used for frequency analysis of signals.
- QFT transforms the computational basis of a qubit to a Fourier Basis.
- Formula of a QFT for Qubit is

$$|\tilde{x}\rangle = QFT\,|x\rangle$$

Fourier basis   Computational basis

$$= \frac{1}{\sqrt{N}} \sum_{y=0}^{N-1} e^{\frac{2\pi i x y}{N}} |y\rangle$$

- Example: QFT for One Qubit

(1) n qubits $\Rightarrow$ 2^n basis states. Define $N \equiv 2^n$. Then,

$$\underset{\text{Fourier basis}}{|\tilde{x}\rangle} \equiv \text{QFT}\, \underset{\text{Computational basis}}{|x\rangle}$$

$$= \frac{1}{\sqrt{N}} \sum_{y=0}^{N-1} e^{\frac{2\pi i x y}{N}} |y\rangle$$

[analogous to inverse discrete Fourier transform]

eg: 1-qubit case [$\Rightarrow N=2$]

$$|\tilde{0}\rangle = \frac{1}{\sqrt{2}} \sum_{y=0}^{1} e^{\frac{2\pi i (0)\cdot y}{2}} |y\rangle = \frac{1}{\sqrt{2}} \sum_{y=0}^{1} |y\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle + |1\rangle\right)$$

$$|\tilde{1}\rangle = \frac{1}{\sqrt{2}} \sum_{y=0}^{1} e^{\frac{2\pi i (1)\cdot y}{2}} |y\rangle = \frac{1}{\sqrt{2}}\left(e^{\frac{2\pi i (0)}{2}}|0\rangle + e^{\frac{2\pi i \cdot 1}{2}}|1\rangle\right)$$

$$= \frac{1}{\sqrt{2}}\left(|0\rangle + (-1)|1\rangle\right)$$

$$= \frac{1}{\sqrt{2}}\left(|0\rangle - |1\rangle\right)$$

- QFT(|0>)= |+>
- QFT(|1>)=|->

Quantum Fourier Transform for n Qubits:

$\frac{x}{4} = 00.x_1 x_2 \cdots x_{n-2}$

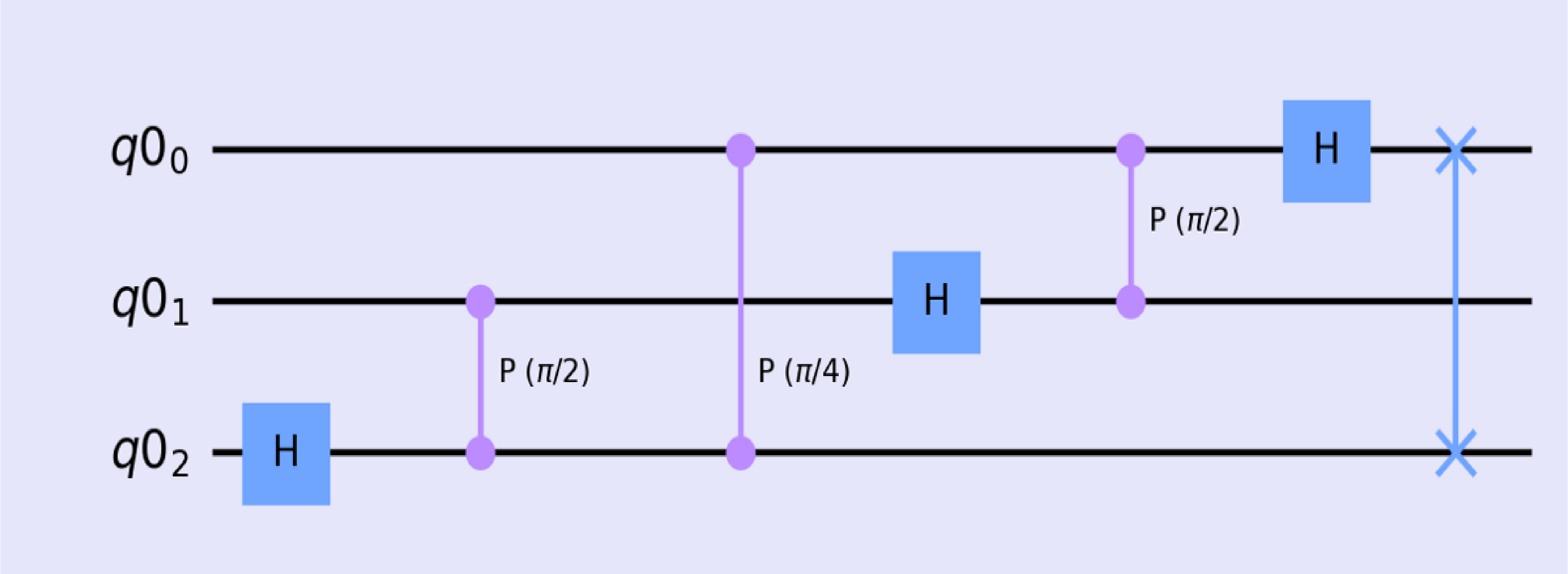
Final form:

$$\frac{1}{\sqrt{N}} \left(|0\rangle + e^{2\pi i \frac{x}{2}} |1\rangle\right) \otimes \left(|0\rangle + e^{2\pi i \frac{x}{4}} |1\rangle\right) \otimes \cdots \left(|0\rangle + e^{2\pi i \frac{x}{2^n}} |1\rangle\right)$$

went from

$|x_1 x_2 \cdots x_n\rangle$ to

Circuit implementation of QFT



Quantum Phase Estimation

- Quantum phase estimation is one of the most important subroutines in quantum computation. It serves as a central building block for many quantum algorithms. The objective of the algorithm is the following:

Given a unitary operator U, the algorithm estimates θ in $U|\psi\rangle = e^{2\pi i\theta}|\psi\rangle$, $|\psi\rangle$ is an eigenvector and $e^{2\pi i\theta}$ is the corresponding eigenvalue. Since U is unitary, all of its eigenvalues have a norm of 1.

Eigen value and Eigen vector

- The result of a measurement of a physical quantity is one of the eigen values of the associated observable.

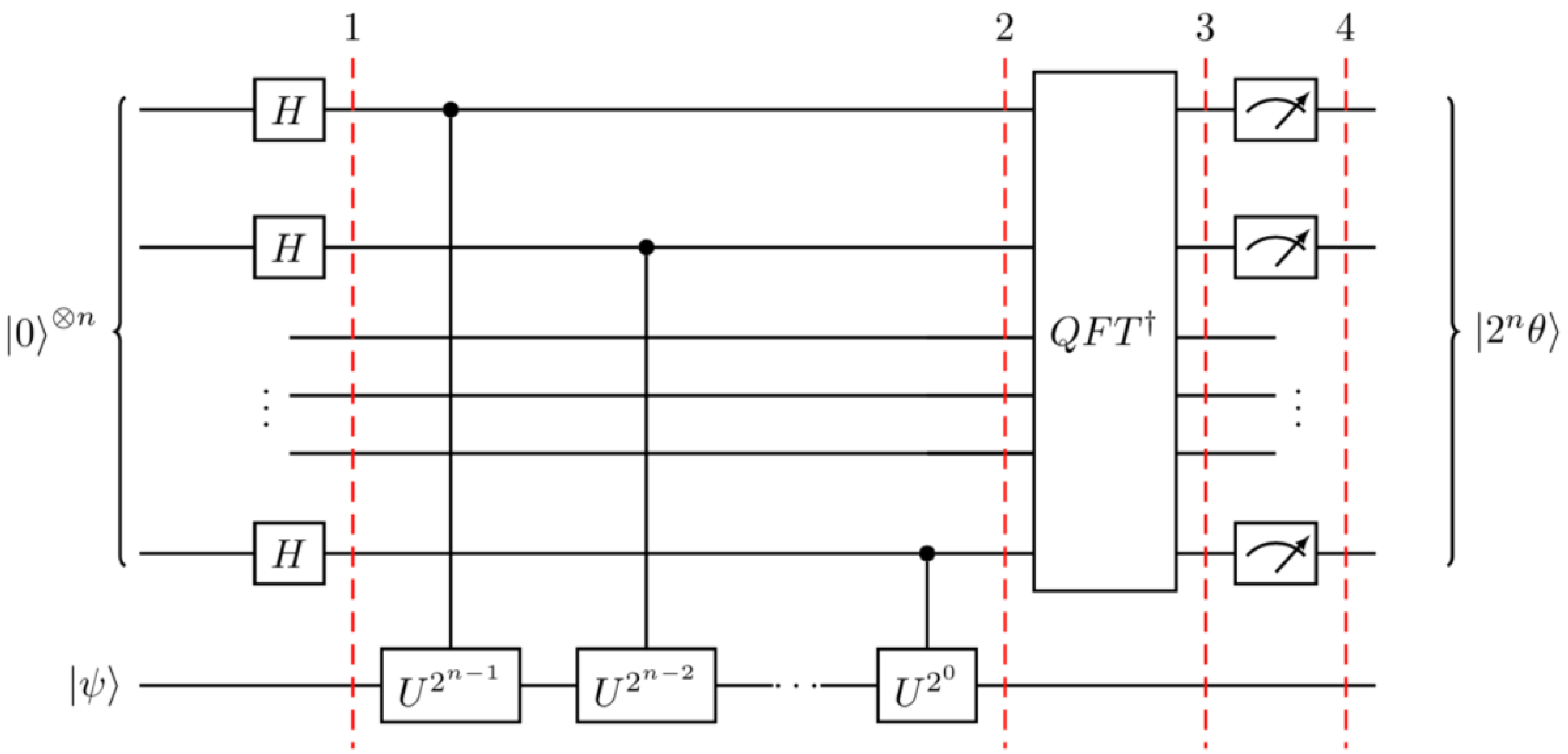
$$A|\varphi> = \lambda|\varphi>$$

Where λ is an eigen value of A, and $|\varphi>$ is an eigenstate or eigen ket.

Controlled Unitary operations

- It applies a Unitary operator U on the target register, only if the corresponding control bit is |1>. U is a unitary with eigenvector $|\varphi>$

such that $U|\varphi> = e^{i2\pi\theta}|\varphi>$.



1
2
3
4
$|0\rangle^{\otimes n}$
H
H
H
$QFT^\dagger$
$|2^n\theta\rangle$
$|\psi\rangle$
$U^{2^{n-1}}$
$U^{2^{n-2}}$
U^{2^0}

- This algorithm uses 2 registers.
- The first register contains m qubits. The more ”m” qubits, the more accurate θ's estimation will be.
- The second register contains |ψ⟩.
- Quantum Phase estimation first prepares the state |0⟩m|ψ⟩ by initializing the first m qubits to |0⟩ and encoding |ψ⟩ in the second register. Hadamard gates are applied to each qubit in the first register
- The state of the control qubit specifies how many times you should be applied the Unitary operator.

$$|\varphi_0\rangle = H^{\otimes m}[|0\rangle]\,|\psi\rangle = \frac{1}{\sqrt{2^m}}[(|0\rangle + |1\rangle)_0\,(|0\rangle + |1\rangle)_1\,(|0\rangle + |1\rangle)_2 \ldots (|0\rangle + |1\rangle)_{2^m-1}]\,|\psi\rangle$$

- 2^{m-1} controlled-U (CU) gates are applied to the second register,
- $U = e^{2\pi i\theta}$

$$|\varphi_1\rangle = cU^{2m-1}|\varphi_0\rangle = \frac{1}{\sqrt{2^m}}[(|0\rangle + e^{2\pi i\theta 2^0}|1\rangle)_0 (|0\rangle + e^{2\pi i\theta}2^1|1\rangle)_1 (|0\rangle + e^{2\pi i\theta}2^2|1\rangle)_2 \ldots (|0\rangle + e^{2\pi i\theta 2^{m-1}}|1\rangle)_{2^m-1}]|\psi\rangle$$

- When we use a Qubit to control the “U-Gate” ,the qubit will turn (due to kickback) propostionally to the phase $e^{2\pi i\theta}$, we can successively apply Cu-gates to repeat this rotation an appropriate number of times until we have encoded the “θ” as a number between 0 and 2^m in the fourier basis.

- We can write the previous equation as,

$$|\varphi_1\rangle = \frac{1}{\sqrt{2^m}}\left[\sum_{k=0}^{2^m-1} e^{2\pi i\theta k}|k\rangle\right]|\psi\rangle$$

- Which is similar to the Fourier Transform,

$$\mathrm{QFT}|x\rangle = \frac{1}{\sqrt{2^m}}\sum_{k=0}^{2^m-1} e^{\frac{2\pi i x k}{2^n}}|k\rangle$$

- The first register of $|\varphi 1>$ is similar to QFT|x> (QFT |θ⟩). To get |θ⟩, the inverse of the Quantum Fourier transform (QFT in reverse) is applied to the first register.

$$\mathrm{QFT}^{-1}\mathrm{QFT}|\theta\rangle = \mathrm{QFT}^{-1}\frac{1}{\sqrt{2^m}}\sum_{k=0}^{2^m-1} e^{2\pi i\theta k}|k\rangle = |\theta_0\,\theta_1\,\theta_2 \ldots\,\theta_m\rangle$$

- The state of the second register doesn't change during computation, so the final state of the system before measurement is $|\theta_0\theta_1\theta_2 \ldots \theta_m >$. Measurement of the first register will result in an approximation of θ.

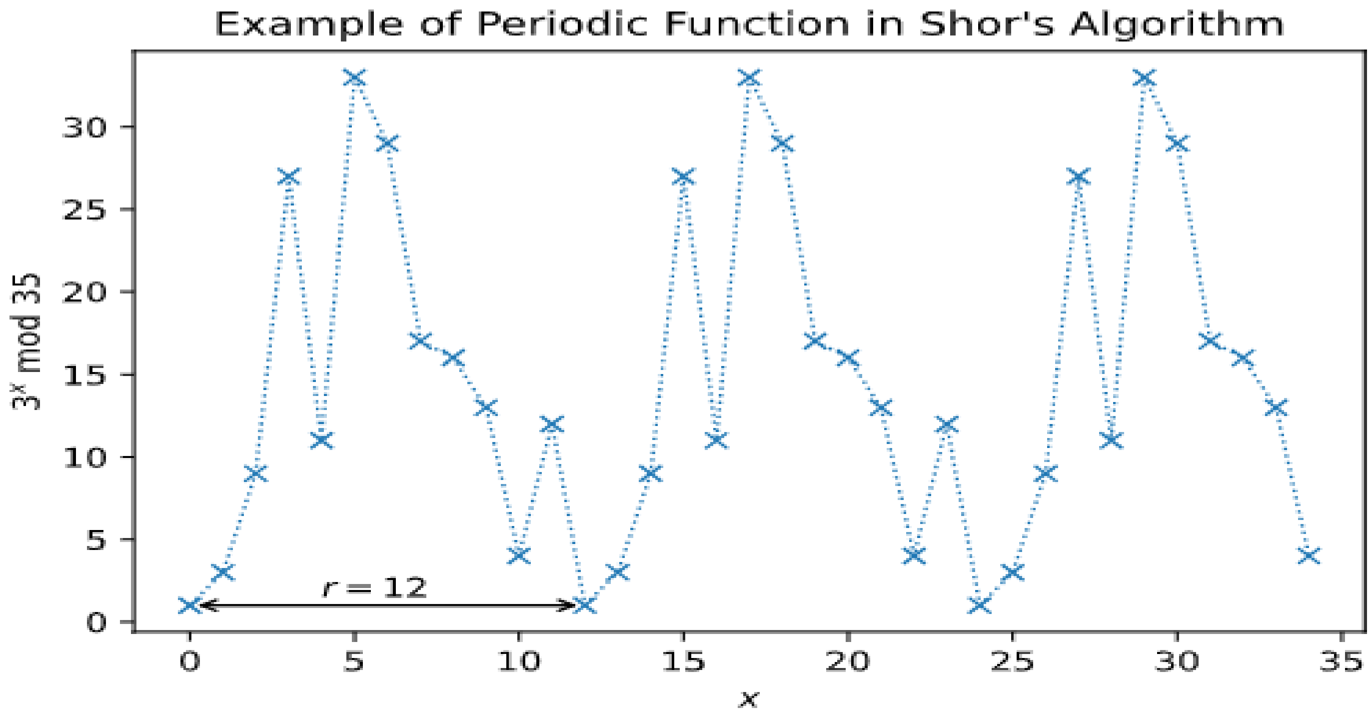
1.	$\|0\rangle\|1\rangle$	initial state
2.	$\rightarrow \frac{1}{\sqrt{2^t}}\sum_{j=0}^{2^t-1}\|j\rangle\|1\rangle$	create superposition
3.	$\rightarrow \frac{1}{\sqrt{2^t}}\sum_{j=0}^{2^t-1}\|j\rangle\|x^j \bmod N\rangle$	apply $U_{x,N}$
	$\approx \frac{1}{\sqrt{r2^t}}\sum_{s=0}^{r-1}\sum_{j=0}^{2^t-1} e^{2\pi isj/r}\|j\rangle\|u_s\rangle$	
4.	$\rightarrow \frac{1}{\sqrt{r}}\sum_{s=0}^{r-1}\|\widetilde{s/r}\rangle\|u_s\rangle$	apply inverse Fourier transform to first register
5.	$\rightarrow \widetilde{s/r}$	measure first register
6.	$\rightarrow r$	apply continued fractions algorithm

Shor’s Algorithm

- It is for factoring integers in polynomial time.(classical algorithm required super polynomial time).
- Shor’s Algorithm uses QPE(Quantum phase estimation) for factoring and period finding.
- Shor’s algorithm, which actually solves the problem of *period finding*. Since a factoring problem can be turned into a period finding problem in polynomial time, an efficient period finding algorithm can be used to factor integers efficiently too.
- The first thing we need to know in order to do Shor’s algorithm is “order finding” .

- f(x)=a^x modN is a periodic function, where ”a” and ”N”are positive integers, a is less than N, and they have no common factors. The period, or order (r), is the smallest (non-zero) integer such that:

$$a^r \ mod \ N = 1$$



Protocols for Shor’s algorithm

1. Pick a number “a” that is co-prime(gcd(N,a)=1) with the Number N(the number we want to factor).
2. Find the “order”(period) r of the function $a^r (mod\ N)$,
3. If r is even :

   then ,$x \equiv a^{r/2}(mod\ N)$,

   if $x + 1 \neq 0(mod\ N)$ then,

   $\{p, q\} = \{\gcd(x + 1, N), \gcd(x - 1, N)\}$

4. Else Find another a.

Example:

- Factoring of 15.

15 = [1111] = 4 bits.

coprime : pick a = 13.

$$\begin{array}{lcccccccc} x = & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 13^x \pmod{15} = & 1, & 13, & 4, & 7, & 1, & 13, & 4, & 7 \end{array}$$

smallest $r > 0$ s.t. $13^r \equiv 1 \pmod{15}$ is $r = 4$.

given $r = 4$,

$$x \equiv 13^{4/2} \pmod{15} \equiv 4 \pmod{15}$$

$$x + 1 = 5 \not\equiv 0 \pmod{15}$$

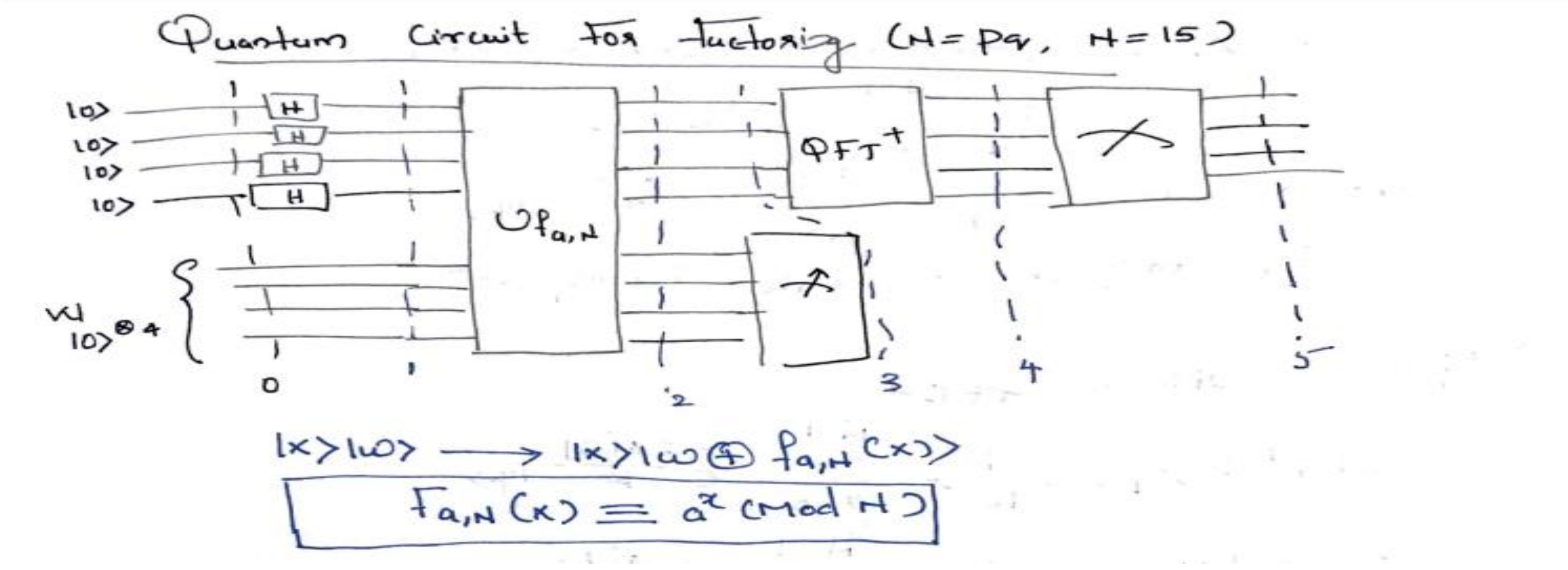
$$\{p, q\} = \{4-1, 4+1\} = \{3, 5\}$$

Quantum Circuit for Factoring N=pq, N=15

Step 1: Determine if the number *N* is a prime, a even number, or an integer power of a prime number. If it is we will not use Shor's algorithm. There are efficient classical methods for determining if a integer *N* belongs to one of the above groups, and providing factors for it if it is. This step would be performed on a classical computer.

Step 2: Pick a random integer *x* that is co-prime to *N*. When two numbers are co-prime it means that their greatest common divisor is 1. There are efficient classical methods for picking such an *x*. This step would be done on a classical computer.

- Step 3: Create a quantum register and partition it into two parts, register 1 and register 2. Thus the state of our quantum computer can be given by: |reg1, reg2 > . Register 1 must have enough qubits to represent N. Register 2 must have enough qubits to represent integers as large as *N* - 1. The calculations for how many qubits are needed would be done on a classical computer.

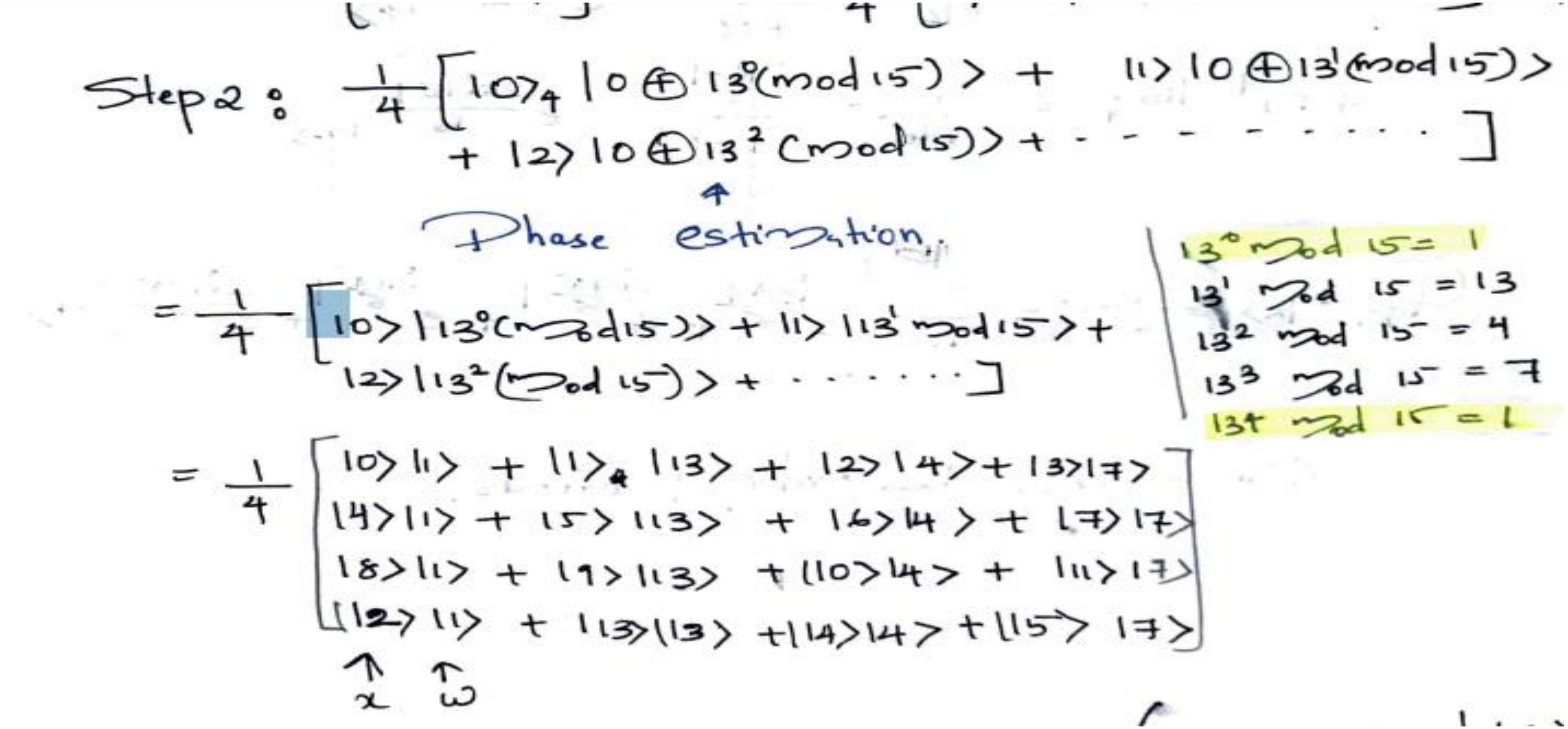


- **Step 4**: Load register 1 with an equally weighted superposition of all integer. Load register 2 with all zeros. This operation would be performed by our quantum computer.

Step 0 : $\underset{x}{|0\rangle^{\otimes 4}} \ \underset{u}{|0\rangle^{\otimes 4}}$

Step 1 : $\left[H^{\otimes 4}|0\rangle\right] |0\rangle^{\otimes 4} = \frac{1}{4}\left[|0\rangle + |1\rangle + |2\rangle + \cdots + |15\rangle\right] |0\rangle_4$

- Step 5: Now apply the transformation x^a mod N to for each number stored in register 1 and store the result in register 2. Due to quantum parallelism this will take only one step.



- Step 6: Measure the second register W.

Step 3: After measuring the "W" register ($|1\rangle, |13\rangle, |4\rangle$, and $|7\rangle$ have equal probability.

If we measure $|7\rangle$ then the $|x\rangle$ become.

$$|x\rangle = \frac{1}{2}\left[|3\rangle, |7\rangle, |11\rangle, |15\rangle\right] \otimes |7\rangle$$

- Step 7: Now compute the discrete Fourier transform on register one.

Step 4: Apply QFT^{+} on the $|x\rangle$ Register.

$$QFT|x\rangle = |\tilde{x}\rangle = \frac{1}{\sqrt{N}} \sum_{y=0}^{N-1} e^{\frac{2\pi i x y}{N}} |y\rangle$$

$$QFT^{+}|x\rangle = |x\rangle = \frac{1}{\sqrt{N}} \sum_{y=0}^{N-1} e^{-\frac{2\pi i x y}{N}} |y\rangle$$

[+ complex conjugate transpose]

Step 4: Apply QFT^{*}

$$QFT^{+}|3\rangle_4 = \frac{1}{\sqrt{16}} \sum_{y=0}^{N-1} e^{-\frac{2\pi i 3y}{16}} |y\rangle$$

$$QFT^{+}|7\rangle_4 = \frac{1}{\sqrt{16}} \sum_{y=0}^{N-1} e^{-\frac{2\pi i 7y}{16}} |y\rangle$$

$$QFT^{+}|11\rangle_4 = \frac{1}{\sqrt{16}} \sum_{y=0}^{N-1} e^{-\frac{2\pi i 11y}{16}} |y\rangle$$

$$QFT^{+}|15\rangle_4 = \frac{1}{\sqrt{16}} \sum_{y=0}^{N-1} e^{-\frac{2\pi i 15y}{16}} |y\rangle$$

- This step is performed by the quantum computer in one step through quantum parallelism. After the discrete Fourier transform our register is in the state:

$$\varphi_{FT}^{+}|x\rangle = \frac{1}{2}\frac{1}{\sqrt{16}}\sum_{y=0}^{15}\left[e^{\frac{-i3\pi y}{8}} + e^{\frac{-i7\pi}{8}y} + e^{\frac{-i11\pi}{8}y} + e^{\frac{-i15\pi}{8}}\right]|y\rangle$$

It will shows the order/Periods.

- Step 8: Measure the register 1.

In this example we will get the states [0,4,8,12] with equal probability. When we measured we will only get one of the Numbers.

- Step 9: Final step, Remaining steps on classical post processing.

The measurement result peak near $J\frac{N}{r}$ or some integer $j \in z, N = 2^n$.

Eg: Measure $|4\rangle$

$j\frac{16}{r} = 4 \rightarrow$ true if $j = 1$ & $r = 4$.

$r \rightarrow$ is the Period.

(1) r is even. then

(2) $x = a^{r/2} \bmod N = 13^{4/2} \pmod{15}$

$= 13^2 \pmod{15} = 4$

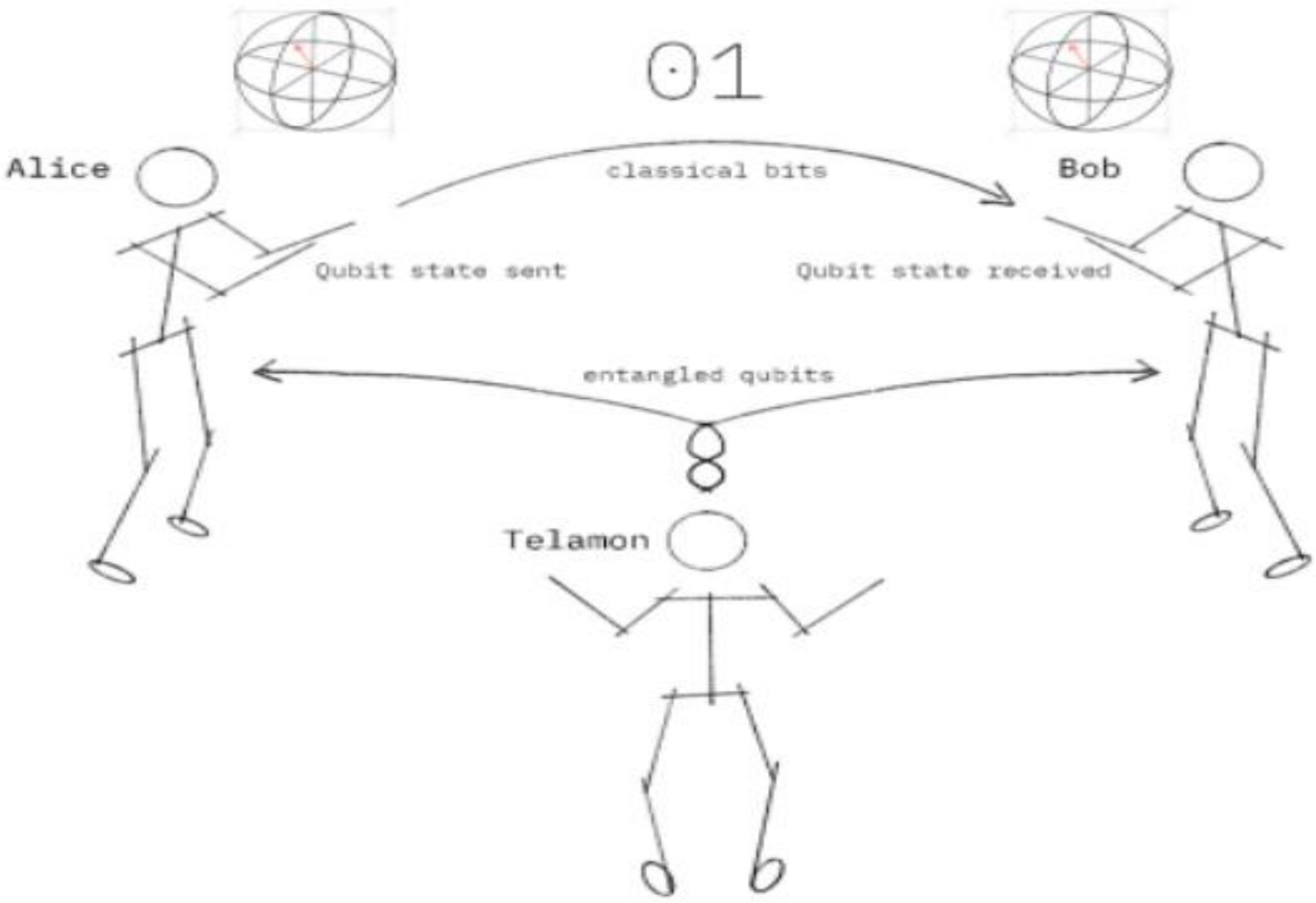
$x+1 = 5$

$x-1 = 3$

$\left.\begin{array}{l} \gcd(x+1, N) = \gcd(5, 15) = 5 \\ \gcd(x-1, N) = \gcd(3, 15) = 3 \end{array}\right\} \rightarrow$ Factors.

Quantum Teleportation

- Teleportation is a procedure that allows one party(Alice)to send a quantum state to her friend(Bob) without that state being transmitted in the usual sense.
- By using Entaglement , Alice and Bob can set up a quantum communication channel that links them together in a quantum way via the EPR paradox.

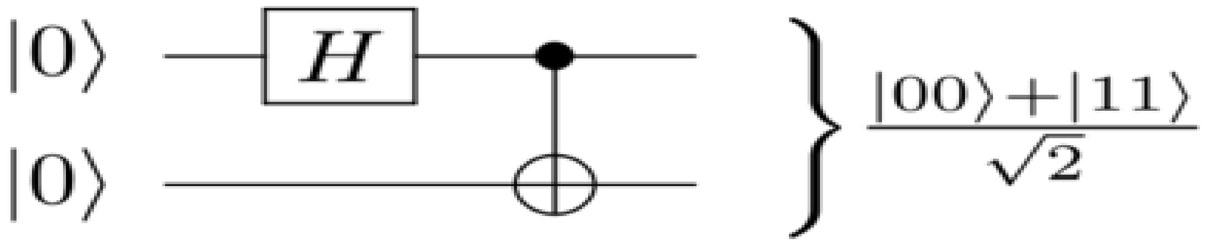


01
Alice
classical bits
Bob
Qubit state sent
Qubit state received
entangled qubits
Telamon

Step 1:

- Quantum Teleportation begins with the fact that Alice needs to transmit $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ (a random qubit) to Bob.
- She doesn't know the state of the qubit. For this, Alice and Bob take the help of a third party (Telamon).
- Telamon prepares a pair of entangled qubits for Alice and Bob

- Creating Entagled Pair:



$$CNOT(H(|0>,|1>) = \frac{1}{\sqrt{2}}(|01> +|10>)$$

$$CNOT(H(|1>,|0>) = \frac{1}{\sqrt{2}}(|00 \succ -|11>)$$

$$CNOT(H(|1>,|1>) = \frac{1}{\sqrt{2}}(|01 \succ -|10>)$$

- Alice and Bob each possess one qubit of the entagled pair(A,B).

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

- This create a three qubit quantum system where Alice has first two qubits and Bob the last one.

$$\begin{aligned}|\psi\rangle \otimes |e\rangle &= \frac{1}{\sqrt{2}}(\alpha|0\rangle \otimes (|00\rangle + |11\rangle) + \beta|1\rangle \otimes (|00\rangle + |11\rangle)) \\ &= \frac{1}{\sqrt{2}}(\alpha|000\rangle + \alpha|011\rangle + \beta|100\rangle + \beta|111\rangle)\end{aligned}$$

Step 2:

- Alice applying CNOT gate on her two qubit.

$$|\psi\rangle \otimes |e\rangle = \frac{1}{\sqrt{2}}(\alpha|0\rangle \otimes (|00\rangle + |11\rangle) + \beta|1\rangle \otimes (|00\rangle + |11\rangle))$$
$$= \frac{1}{\sqrt{2}}(\alpha|000\rangle + \alpha|011\rangle + \beta|100\rangle + \beta|111\rangle)$$

$$\text{Applying CNOT gate} = \frac{1}{\sqrt{2}}(\alpha|000 > +\alpha|011 > +\beta|110 > +\beta|101 >)$$

Step 3:

- Alice applying Hadamard gate to her first qubit.

$$|\psi\rangle \otimes |e\rangle = \frac{1}{\sqrt{2}}(\alpha|0\rangle \otimes (|00\rangle + |11\rangle) + \beta|1\rangle \otimes (|00\rangle + |11\rangle))$$
$$= \frac{1}{\sqrt{2}}(\alpha|000\rangle + \alpha|011\rangle + \beta|100\rangle + \beta|111\rangle)$$

$$\text{Applying CNOT gate} = \frac{1}{\sqrt{2}}(\alpha|000> +\alpha|011> +\beta|110> +\beta|101>)$$

$$Applying\ Hadamard\ Gate =$$
$$\frac{\alpha(|0> +|1>)}{\sqrt{2}}\frac{(|00> +|11>}{\sqrt{2}} + \frac{\beta(|0> -|1>)}{\sqrt{2}}\frac{(|10> +|01>}{\sqrt{2}}$$

- $= \frac{1}{2}\big(\alpha(|000 > +|011 > +|100 > +|111 >) + \beta(|010 \succ +|001 \succ$

$$
\begin{aligned}
= \frac{1}{2}\big(\quad & |00\rangle(\alpha|0\rangle + \beta|1\rangle) \\
& + |01\rangle(\alpha|1\rangle + \beta|0\rangle) \\
& + |10\rangle(\alpha|0\rangle - \beta|1\rangle) \\
& + |11\rangle(\alpha|1\rangle - \beta|0\rangle) \quad \big)
\end{aligned}
$$

Step 4:

- Alice measure the first two qubit and send them as two classical bits to Bob.
- The result Alice obtain is always one of the 4 standard basis states |00>,|01>,|10> and |11> with equal probability.
- On the basis of Alice measurement Bob's state will be projected to,

$$
\begin{aligned}
|00\rangle &\rightarrow (\alpha|0\rangle + \beta|1\rangle) \\
|01\rangle &\rightarrow (\alpha|1\rangle + \beta|0\rangle) \\
|10\rangle &\rightarrow (\alpha|0\rangle - \beta|1\rangle) \\
|11\rangle &\rightarrow (\alpha|1\rangle - \beta|0\rangle)
\end{aligned}
$$

Step 5

- Bob, on receiving the bits from Alice, knows he can obtain the original state $|\psi\rangle$ by applying appropriate transformations on his qubit that was once part of the entangled pair.
- The transformations Bob needs to apply are:

Bob's State	Bits Received	Gate Applied
$(\alpha\|0\rangle + \beta\|1\rangle)$	00	I
$(\alpha\|1\rangle + \beta\|0\rangle)$	01	X
$(\alpha\|0\rangle - \beta\|1\rangle)$	10	Z
$(\alpha\|1\rangle - \beta\|0\rangle)$	11	ZX

- After this step Bob will have successfully reconstructed Alice's state.