Part VI

Looking into the future

Previous lecture

- Set up mathematics related to qubits for both single qubit state and two qubit states.
- Introduced single qubit gates and wrote matrices for them.
- Introduced two qubit gates and discussed C-NOT gate.
- A little inspection shows that the matrices corresponding to different gates are combinations of Pauli matrices.
- Pauli matrices satisfy a desirable property called the unitarity.

Lecture 37

Introduction to Quantum Computing-I
(Quantum Parallelism)

Unitarity of a matrix and why it is needed

- A wavefunction must remain normalized after an operation is carried out on it.
- What would happen if the wavefunction did not remain normalized?
- That would mean that system is becoming smaller or part of it is getting lost. *In this connection, recall when a particle got transmitted across a barrier, the amplitude of the transmitted and reflected wave are smaller than that of the incident wave.*
- Normalization of wavefunction is assured if the operation is represented by a unitary matrix. This is the of a unitary matrix.
- If an operation is not represented by a unitary matrix, the result of that would be system losing some of its parts.

Mathematical definition of a unitary matrix

- Recall the definition of the adjoint of a matrix

$$\begin{pmatrix} A_{11} \\ A_{21} \end{pmatrix}^{\dagger} = \begin{pmatrix} A_{11} \\ A_{21} \end{pmatrix}^{T*} = \begin{pmatrix} A_{11}^* & A_{21}^* \end{pmatrix} \quad ; \quad \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix}^{\dagger} = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix}^{T*} = \begin{pmatrix} A_{11}^* & A_{21}^* \\ A_{12}^* & A_{22}^* \end{pmatrix}$$

- And normalization of a single qubit state

$$\begin{pmatrix} \alpha^* & \beta^* \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = 1$$

- Now suppose after an operation (passing through a gate), a qubit transforms as follows

$$\begin{pmatrix} \alpha \\ \beta \end{pmatrix} \rightarrow \begin{pmatrix} \overline{\alpha} \\ \overline{\beta} \end{pmatrix} = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix}$$

Mathematical definition of a unitary matrix

- If the transformed qubit remains normalized then

$$(\overline{\boldsymbol{\alpha}}^* \quad \overline{\boldsymbol{\beta}}^*)\begin{pmatrix}\overline{\boldsymbol{\alpha}}\\ \overline{\boldsymbol{\beta}}\end{pmatrix} = (\boldsymbol{\alpha}^* \quad \boldsymbol{\beta}^*)\begin{pmatrix}\boldsymbol{A_{11}} & \boldsymbol{A_{12}}\\ \boldsymbol{A_{21}} & \boldsymbol{A_{22}}\end{pmatrix}^{\boldsymbol{T*}}\begin{pmatrix}\boldsymbol{A_{11}} & \boldsymbol{A_{12}}\\ \boldsymbol{A_{21}} & \boldsymbol{A_{22}}\end{pmatrix}\begin{pmatrix}\boldsymbol{\alpha}\\ \boldsymbol{\beta}\end{pmatrix} = \mathbf{1}$$

- This means

$$\begin{pmatrix}\boldsymbol{A_{11}} & \boldsymbol{A_{12}}\\ \boldsymbol{A_{21}} & \boldsymbol{A_{22}}\end{pmatrix}^{\dagger}\begin{pmatrix}\boldsymbol{A_{11}} & \boldsymbol{A_{12}}\\ \boldsymbol{A_{21}} & \boldsymbol{A_{22}}\end{pmatrix} = \begin{pmatrix}\mathbf{1} & \mathbf{0}\\ \mathbf{0} & \mathbf{1}\end{pmatrix} = \mathbf{I}$$

- The definition of a **unitary matrix** U is then

$$\boldsymbol{U^{\dagger}U} = \mathbf{I}$$

- If the matrix is real then it is also called an **orthogonal matrix** and follows

$$\boldsymbol{U^{T}U} = \mathbf{I}$$

Unitary requirement for matrix corresponding to a gate

- If we propose a quantum gate, the corresponding matrix must be unitary.
- It is easily checked that Pauli matrices are unitary matrices.

Quantum parallelism
How does a quantum computer make calculations faster?

Example **(Reference: The Code Book …………… by Simon Singh):**

- Find a number X such that digits appearing in X^2 and X^3 give the digits 0, 1, 2, 3 …….. with each digit appearing only once.

$$\mathbf{13^2 = 169} \qquad \mathbf{13^3 = 2197} \quad (only\ seven\ digits\ appear)$$

$$\mathbf{27^2 = 729} \qquad \mathbf{27^3 = 19683} \quad (only\ seven\ digits\ appear)$$

- Answer

$$\mathbf{69^2 = 4761} \qquad \mathbf{69^3 = 328509} \qquad (all\ digits\ appear)$$

Using a classical computer to solve the problem on the previous slide

- Pick a number n and calculate n^2 and n^3.
- Check all the digits in n^2 and n^3 starting , for example, at $n = 11$ and increasing it by 1 at a time.
- The computer will stop when it reaches 69.
- Suppose each calculation takes 5 seconds, then total time taken will be 5 minutes.

How would one use a quantum computer to solve the same problem?

- Take advantage of the superposition of states.
- In binary the numbers are (we will write number up to 127)

$$|\mathbf{0}\rangle = \mathbf{0000000} \qquad |\mathbf{1}\rangle = \mathbf{0000001}$$

$$|\mathbf{2}\rangle = \mathbf{0000010} \qquad |\mathbf{3}\rangle = \mathbf{0000011}$$

$$\dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots$$

$$\dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots$$

$$|\mathbf{126}\rangle = \mathbf{1111110} \qquad |\mathbf{3}\rangle = \mathbf{1111111}$$

- **Note:** Each of this state will run separately on a classical computer.

Use of a quantum computer to solve..........

- Write each number up to 127 as a 7-qubit state

$$|\mathbf{0}\rangle = |\mathbf{0}\rangle|\mathbf{0}\rangle|\mathbf{0}\rangle|\mathbf{0}\rangle|\mathbf{0}\rangle|\mathbf{0}\rangle|\mathbf{0}\rangle \qquad |\mathbf{1}\rangle = |\mathbf{0}\rangle|\mathbf{0}\rangle|\mathbf{0}\rangle|\mathbf{0}\rangle|\mathbf{0}\rangle|\mathbf{0}\rangle|\mathbf{1}\rangle$$

$$|\mathbf{2}\rangle = |\mathbf{0}\rangle|\mathbf{0}\rangle|\mathbf{0}\rangle|\mathbf{0}\rangle|\mathbf{0}\rangle|\mathbf{1}\rangle|\mathbf{0}\rangle \qquad |\mathbf{3}\rangle = |\mathbf{0}\rangle|\mathbf{0}\rangle|\mathbf{0}\rangle|\mathbf{0}\rangle|\mathbf{0}\rangle|\mathbf{1}\rangle|\mathbf{1}\rangle$$

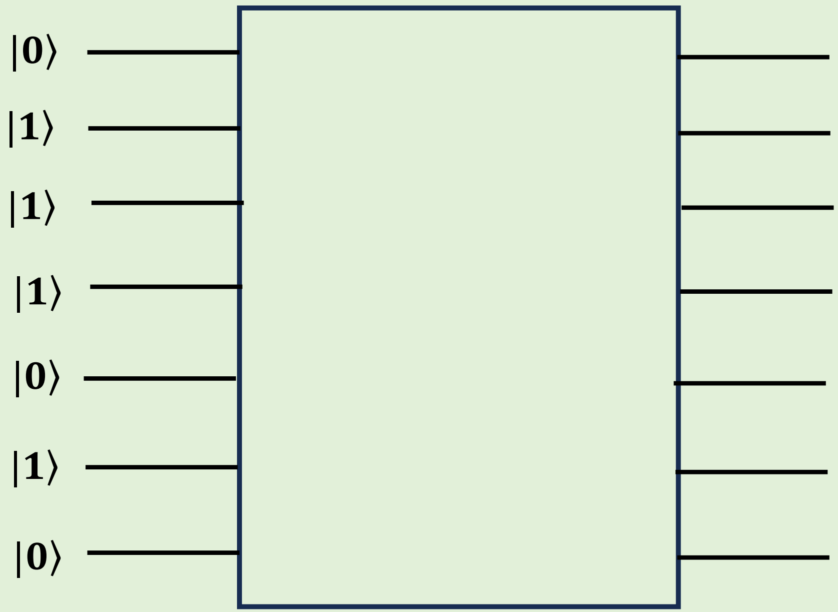
$$\dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots$$

$$\dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots$$

$$|\mathbf{126}\rangle = |\mathbf{1}\rangle|\mathbf{1}\rangle|\mathbf{1}\rangle|\mathbf{1}\rangle|\mathbf{1}\rangle|\mathbf{1}\rangle|\mathbf{0}\rangle \qquad |\mathbf{127}\rangle = |\mathbf{1}\rangle|\mathbf{1}\rangle|\mathbf{1}\rangle|\mathbf{1}\rangle|\mathbf{1}\rangle|\mathbf{1}\rangle|\mathbf{1}\rangle$$

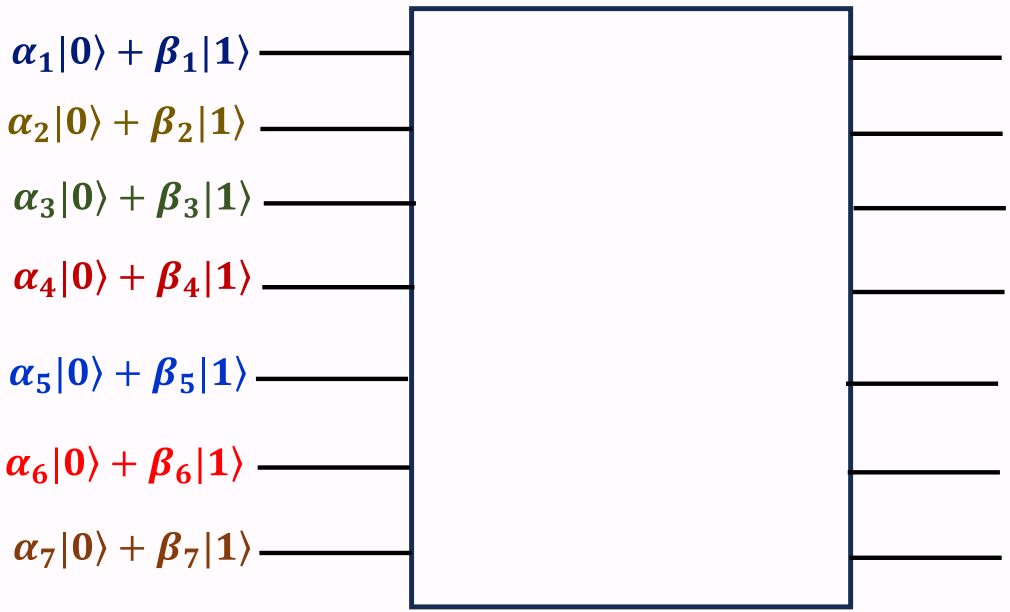
Use of a quantum computer to solve..........

If we run these qubits on a 7-qubit gate as follows, it offers no advantage over a classical computer as each number is processed one at a time



Use of a quantum computer to solve..........

To take advantage of the parallelism offered by superposition of state, each port of the gate is given an input of the form $\boldsymbol{\alpha}|\mathbf{0}\rangle + \boldsymbol{\beta}|\mathbf{1}\rangle$.



Use of a quantum computer to solve..........

- Input to the gate now is the 7-qubit state

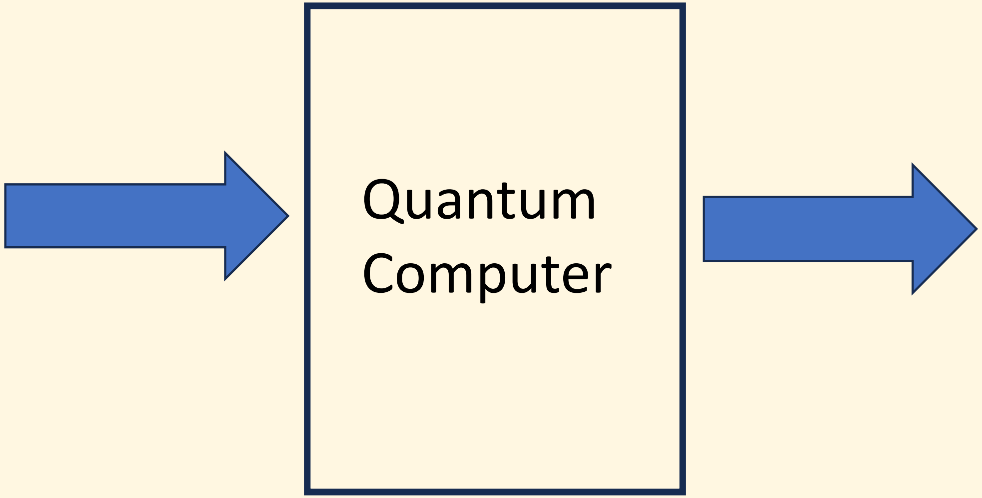
$$|\psi\rangle = \prod_{i=1}^{7} (\alpha_i|0\rangle + \beta_i|1\rangle) = (\alpha_1|0\rangle + \beta_1|1\rangle)(\alpha_2|0\rangle + \beta_2|1\rangle) \ldots \ldots \ldots \ldots$$

$$= \sum_{n=0}^{127} C_n|n\rangle$$

- Now all numbers from 0 to 127 are being given as input to the quantum gate now. These are all processed together.
- The entire calculation will therefore take only 5 seconds.

Input and output of a quantum computer (7-qubit gate) for the problem at hand

$$\sum_{n=0}^{127} C_n |n\rangle$$



A linear combination of answer to each number n

We should now have a mechanism to read all the answers individually and pick the right one.

- This example evidently is an extreme case where one assumed that all possible states could be processed and extracted together. The point that is being made is about the possibility of doing so.
- In reality, may be only some limited computations can be done but whenever it can be done advantage over classical computer will be enormous.
- Doing quantum computation of course requires development of different algorithms and programming in addition to the required hardware, We now demonstrate this through the first such algorithm called the **Deutsch's algorithm**. It was developed by David Deutsch and was significant in starting the field of quantum computing. In this algorithm, a property of a function is evaluated without actually calculating the function.

Deutsch's algorithm

- Define a function f from the domain $\{0,1\}$ to the range $\{0,1\}$
- The input is 0 or 1 and the output is also 0 or 1.
- There are four such functions

$f_0 \quad f_0(0) = 0 \quad f_0(1) = 0$

$f_1 \quad f_1(0) = 0 \quad f_1(1) = 1$

$f_2 \quad f_2(0) = 1 \quad f_2(1) = 0$

$f_3 \quad f_3(0) = 1 \quad f_3(1) = 1$

- Funtions f_0 and f_3 are known as constant functions since output for both 0 and 1 is the same for these two functions.
- Funtions f_1 and f_2 are known as balanced functions since output is equally 0 and 1.

Deutsch’s question

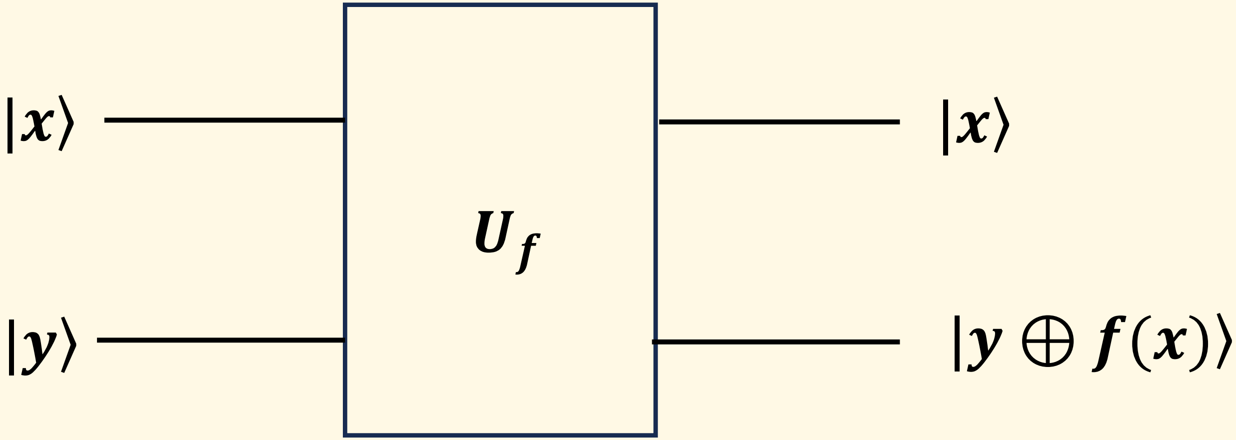
Given one of the functions f_0, f_1, f_2 and f_3 at random, how many evaluations of the function do we have to make to determine whether it is a constant function or a balanced function?

Number of function evaluations with a classical calculation

- Suppose we give input 0, we will get output either 0 or 1.
- If output is 0, the function could be either f_0 (a constant function) or f_1 (a balanced function).
- If output is 1, the function could be either f_2 (a balanced function) or f_3 (a constant function).
- In either case, we will have to do one more evaluation by giving input 1.
- Thus, two evaluations are required to answer the question posed.

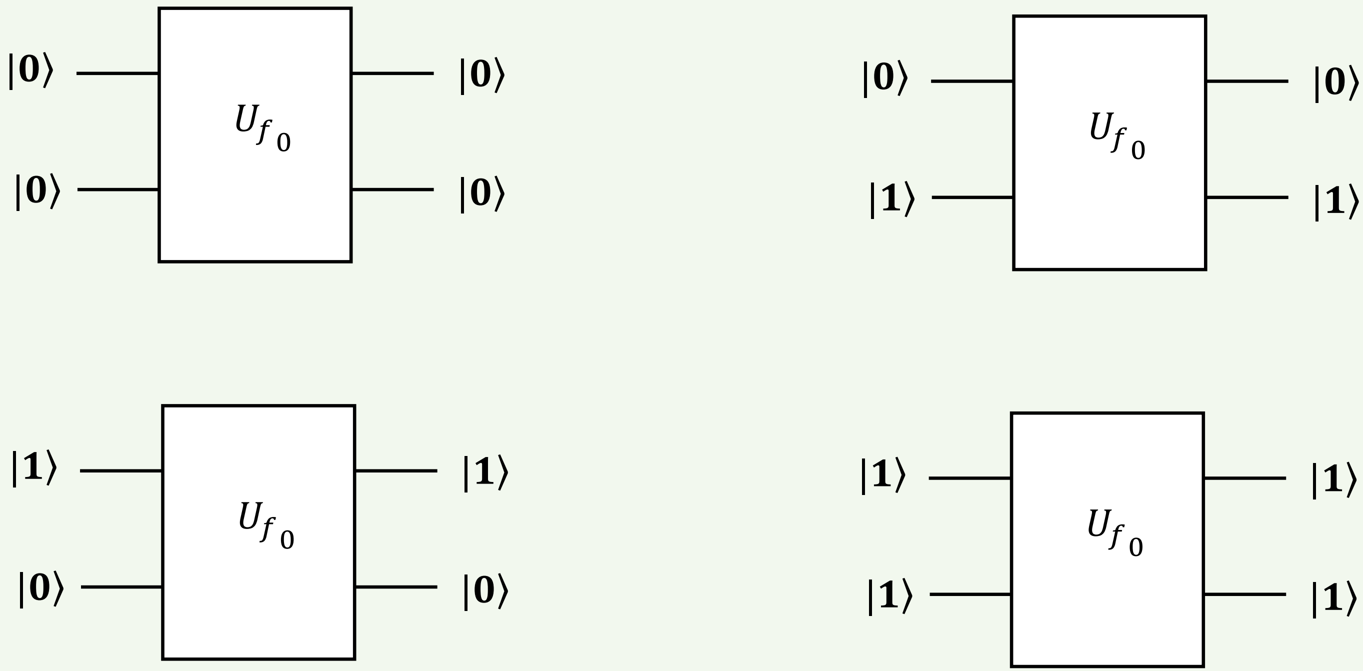
Quantum version of the question

- Make the following gate corresponding to a function f



- Ask the question: Given a gate at random, is it corresponding to a constant or a balanced function?

Operation of U_{f_0}

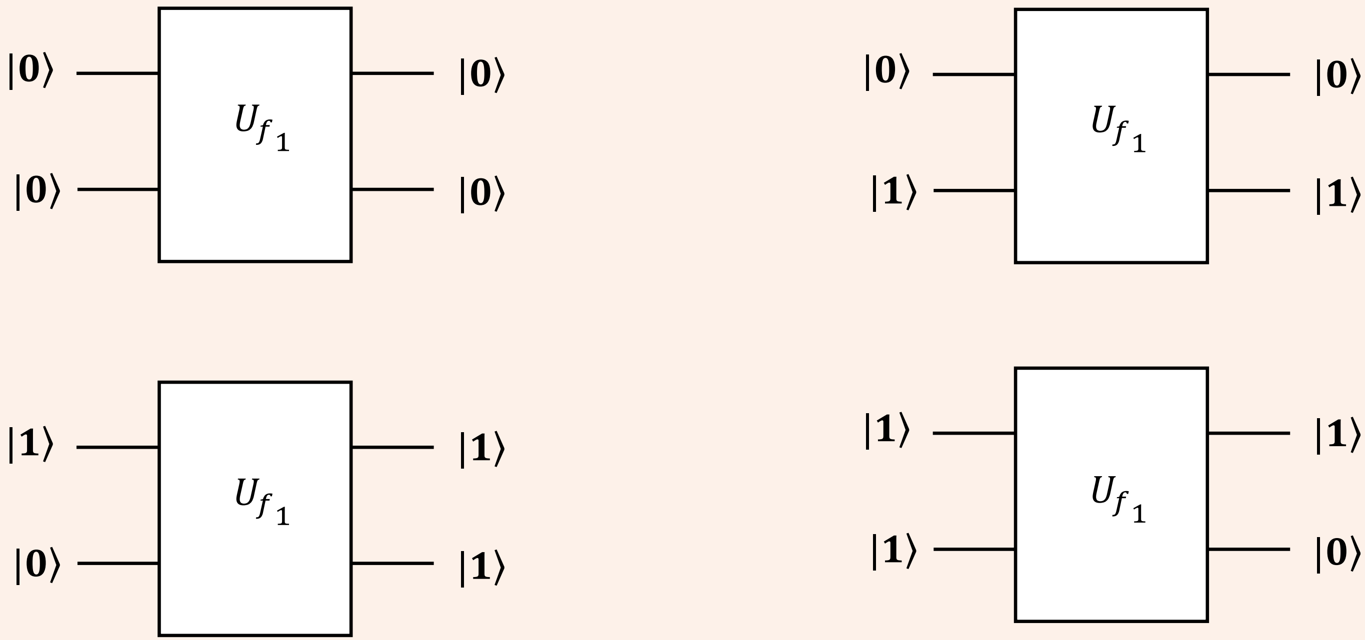


Matrix corresponding to $\boldsymbol{U_{f_0}}$

$$\boldsymbol{U_{f_0}|00\rangle = 1 \times |00\rangle + 0 \times |01\rangle + 0 \times |10\rangle + 0 \times |11\rangle}$$
$$\boldsymbol{U_{f_0}|01\rangle = 0 \times |00\rangle + 1 \times |01\rangle + 0 \times |10\rangle + 0 \times |11\rangle}$$
$$\boldsymbol{U_{f_0}|10\rangle = 0 \times |00\rangle + 0 \times |01\rangle + 1 \times |10\rangle + 0 \times |11\rangle}$$
$$\boldsymbol{U_{f_0}|11\rangle = 0 \times |00\rangle + 0 \times |01\rangle + 0 \times |10\rangle + 1 \times |11\rangle}$$

The corresponding matrix is identity matrix

Operation of $\boldsymbol{U_{f_1}}$



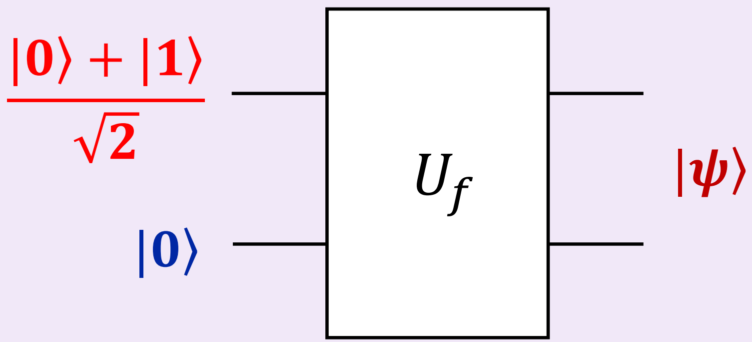
Matrix corresponding to $\boldsymbol{U_{f_1}}$

$$\boldsymbol{U_{f_1}|00\rangle = 1 \times |00\rangle + 0 \times |01\rangle + 0 \times |10\rangle + 0 \times |11\rangle}$$
$$\boldsymbol{U_{f_1}|01\rangle = 0 \times |00\rangle + 1 \times |01\rangle + 0 \times |10\rangle + 0 \times |11\rangle}$$
$$\boldsymbol{U_{f_1}|10\rangle = 0 \times |00\rangle + 0 \times |01\rangle + 0 \times |10\rangle + 1 \times |11\rangle}$$
$$\boldsymbol{U_{f_1}|11\rangle = 0 \times |00\rangle + 0 \times |01\rangle + 1 \times |10\rangle + 0 \times |11\rangle}$$

The corresponding matrix is

$$U_{f_1} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

A Remarkable result by applying $\boldsymbol{U_f}$



Here $|\psi\rangle$ is a two qubit state given as

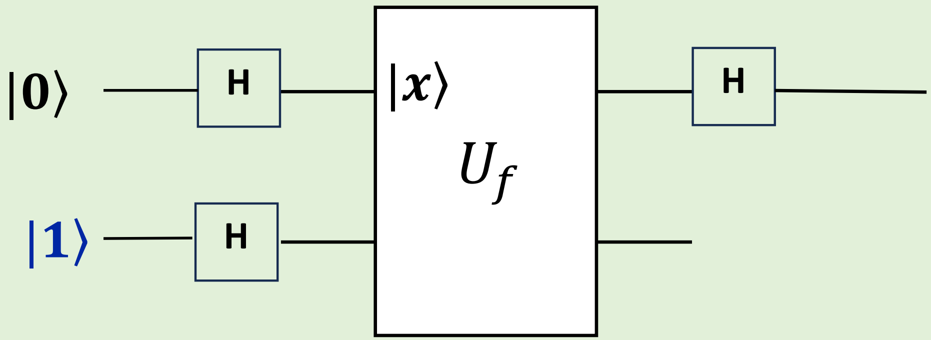
$$|\psi\rangle = \frac{|0\rangle|0\oplus f(0)\rangle + |1\rangle|0\oplus f(1)\rangle}{\sqrt{2}}$$

$$= \frac{|0\rangle|f(0)\rangle + |1\rangle|f(1)\rangle}{\sqrt{2}}$$

The results obtained above is a remarkable result. We have not done separate calculation for $|f(0)\rangle$ and $|f(1)\rangle$. Rather, we have calculated them simultaneously. This is an example of the parallelism offered by a quantum computer. Although how to extract their sum is still not clear. This will be done next using Deutsch's algorithm.

Deutsch's algorithm

- We wish to find whether a gate is corresponding to a constant function or a balanced function.
- Make the following circuit with the inputs shown



Mathematical analysis of the circuit in the slide above

Two qubit state entering U_f is $|\boldsymbol{\psi_1}\rangle = \frac{\mathbf{1}}{\mathbf{2}}\sum_{\boldsymbol{x}=\mathbf{0}}^{\mathbf{1}} |\boldsymbol{x}\rangle(|\mathbf{0}\rangle - |\mathbf{1}\rangle)$

$$U_f|\boldsymbol{\psi_1}\rangle = \frac{\mathbf{1}}{\mathbf{2}}\sum_{x=0}^{x=1}(|\boldsymbol{x}\rangle|\mathbf{0}\oplus\boldsymbol{f}(\boldsymbol{x})\rangle - |\boldsymbol{x}\rangle|\mathbf{1}\oplus\boldsymbol{f}(\boldsymbol{x})\rangle)$$

It is clear if

$f(x) = 0$ then $(|\boldsymbol{x}\rangle|\mathbf{0}\oplus\boldsymbol{f}(\boldsymbol{x})\rangle - |\boldsymbol{x}\rangle|\mathbf{1}\oplus\boldsymbol{f}(\boldsymbol{x})\rangle) = \frac{\mathbf{1}}{\mathbf{2}}|\boldsymbol{x}\rangle\,(|\mathbf{0}\rangle - |\mathbf{1}\rangle)$

$f(x) = 1$ then $(|\boldsymbol{x}\rangle|\mathbf{0}\oplus\boldsymbol{f}(\boldsymbol{x})\rangle - |\boldsymbol{x}\rangle|\mathbf{1}\oplus\boldsymbol{f}(\boldsymbol{x})\rangle) = -\frac{\mathbf{1}}{\mathbf{2}}|\boldsymbol{x}\rangle\,(|\mathbf{0}\rangle - |\mathbf{1}\rangle)$

This gives:

$$(|\boldsymbol{x}\rangle|\mathbf{0}\oplus\boldsymbol{f}(\boldsymbol{x})\rangle - |\boldsymbol{x}\rangle|\mathbf{1}\oplus\boldsymbol{f}(\boldsymbol{x})\rangle) = (-\mathbf{1})^{\boldsymbol{f}(\boldsymbol{x})}\frac{\mathbf{1}}{\mathbf{2}}|\boldsymbol{x}\rangle\,(|\mathbf{0}\rangle - |\mathbf{1}\rangle)$$

Mathematical analysis ....

- So,

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

$$= \frac{1}{2}\{(-1)^{f(0)}|0\rangle + (-1)^{f(1)}|1\rangle\}(|0\rangle - |1\rangle)$$

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

- Finally the first state $\frac{1}{\sqrt{2}}\{(-1)^{f(0)}|0\rangle + (-1)^{f(1)}|1\rangle\}$ is put through a Hadamard gate.

Mathematical analysis ....

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

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

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

- If the function is a constant function, $(-1)^{f(0)}-(-1)^{f(1)}=0$ and if it is a balanced function $(-1)^{f(0)}+(-1)^{f(1)}=0$.
- So the output from the final H gate is $\pm|0\rangle$ if the function is a constant and $\pm|1\rangle$ if the function is balanced

Conclusion

- If $\boldsymbol{f}(\mathbf{0}) = \boldsymbol{f}(\mathbf{1})$, the qubit in x-port is $|\mathbf{0}\rangle$ and if $\boldsymbol{f}(\mathbf{0}) \neq \boldsymbol{f}(\mathbf{1})$, the qubit in x-port is $|\mathbf{1}\rangle$. This result is can also be expressed as $|\boldsymbol{f}(\mathbf{0}) \oplus \boldsymbol{f}(\mathbf{1})\rangle$.
- The final result is that we know the relationship between $\boldsymbol{f}(\mathbf{0})$ and $\boldsymbol{f}(\mathbf{1})$ without calculating them separately. This is a demonstration that a quantum computer requires less number of steps for calculation and is therefore faster than a classical computer.