Suppose \( f(x); \exists 0,1 \Rightarrow \exists 0,1 \) is a function of a one-bit domain and range.

To compute this function on a quantum computer, we consider a 2-qubit \( \mathcal{Q} \) which starts in state \( \ket{0, y} \).

With an appropriate sequence of logic gates, it is possible to transform this state in \( \ket{0, y \oplus f(x)} \), where \( \oplus \) is addition modulo 2.

\[
\begin{array}{c}
\text{data register} \\
\ket{0, y} \\
\text{target register} \\
\ket{0, y} \\
\end{array}
\]

\[
\begin{array}{c}
\text{if } \oplus \text{ unite } \\
\text{if } y = 0, \text{ the final state of the second} \\
\text{qubit is } f(x). \text{ Indeed} \\
\end{array}
\]

\[
\begin{array}{c|c|c|c}
\hline
y & f(x) & y \oplus f(x) \\
\hline
0 & 0 & 0 \\
0 & 1 & 1 \\
1 & 0 & 1 \\
1 & 1 & 0 \\
\hline
\end{array}
\]

Can we use \( y \) for evaluate \( f(0) \) \& \( f(1) \) simultaneously.
\[ U_f \frac{10\rangle + 11\rangle}{\sqrt{2}} \otimes 10\rangle = \left\{ \text{10, } f(0) \rangle + 11, f(1) \rangle \right\} \frac{1}{\sqrt{2}} = 14\rangle \]

This is a remarkable state! The different terms contain information about \( f(0) \) and \( f(1) \). It is as if we have evaluated \( f(x) \) for two values of \( x \) simultaneously, a feature known as quantum parallelism.

A single \( U_f \) gate is used to evaluate the function for different \( x \) simultaneously!

How do we generalize the process to an arbitrary number of bits? We use the Hadamard transform! This operation is just \( n \) Hadamard gates acting in parallel on \( n \) qubits.

For \( n = 2 \)

\[ \begin{array}{c}
H \\
H
\end{array} \]
\[ |\psi\rangle = \frac{1}{\sqrt{12}} \left[ |0\rangle + |1\rangle \right] \otimes \frac{1}{\sqrt{12}} \left[ |0\rangle + |1\rangle \right] = \frac{1}{2} \left[ |00\rangle + |01\rangle + |10\rangle + |11\rangle \right] \]

We write \( H^\otimes 2 \) the parallel action of the 2 Hadamard gates.

More generally, \( |0\rangle \otimes |0\rangle \rightarrow H \otimes H \)

\[
\rightarrow \frac{1}{\sqrt{2^m}} \sum_{x} |x\rangle = \left[ 2^m \text{ states} \right]^{\text{superposition}} \text{[result in gate]}
\]

\( \Sigma \) is over all possible values of \( x \).

The Hadamard transform produces an equal superposition of all computational basis states in \( \mathbb{C}^m \).

Quantum parallel evaluation of a function with an \( m \) bit input \( x \) and 1 bit output, \( f(x) \), can then be performed as follows:

Prepare the \( n+1 \) qubit state \( |0\rangle^\otimes n |1\rangle \)

Apply \( H \) to first \( n \) qubit with \( U_f \)

\[
|\psi\rangle = \frac{1}{\sqrt{2^m}} \sum_{x} |x\rangle \overset{f(x)}{\rightarrow} H \otimes H |\psi\rangle
\]

\( \triangleright \) Measurement on \( |\psi\rangle \) will give \( f(x) \) for a single value of \( x \). Each term in the sum has an equal probability.
\[ H|0\rangle = \frac{1}{\sqrt{2}} (|0\rangle + 1|1\rangle) \]

\[ (H \otimes H \otimes \cdots \otimes H)|00\ldots0\rangle = \frac{1}{\sqrt{2^n}} (|0\rangle + 1|1\rangle)(|0\rangle + 1|1\rangle)\cdots(|0\rangle + 1|1\rangle) \]

\[ = \frac{1}{\sqrt{2^n}} \sum_{\alpha=0}^{2^n-1} |\alpha\rangle \]

\[ |0,0,\ldots,1,\ldots0\rangle \]

all possible combinations
How do we extract information about more than one value of \( f(x) \) from superposition states like \( \frac{1}{\sqrt{2}} |1, e, f(e) \rangle \)

Deutsch's algorithm

Suppose Alice has 2 bits, 0, 1, she can send to Bob. Bob evaluates some Boolean function of the bit sent by Alice. What are the results that Bob can get?

<table>
<thead>
<tr>
<th>Alice</th>
<th>f1</th>
<th>f2</th>
<th>f3</th>
<th>f4</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

Only 4 possibilities.

The two functions can be regrouped in two subsets. \( f_1, f_4 \) give the constant value 0 or 1 for the two bits 0 & 1 sent by Alice. \( f_2, f_3 \) gives either 0 or 1 after Alice sends her 2 bits. These are referred to as balanced functions.

Question: Classically, can Alice send either 0 or 1 and find out if Bob selected a constant or balanced function to evaluate \( f(x) \)?

Answer: No, Alice must send both 0 & 1 before being able to answer that question.
If we can find a circuit which can compute \( f_i(0) \oplus f_i(1) \) in one result, Alice will be able to answer the question of a constant or balanced function used by Bob in one result.

How can we use UP to do this?

\[
x = \left[ \begin{array}{c|c}
x_1 & x_2 \\
\end{array} \right] \\
y = \left[ \begin{array}{c|c}
y_1 & y_2 \\
\end{array} \right] \\
\]

\[
\begin{align*}
(x_1, x_2) = & \left[ \begin{array}{c|c}
0 & 0 \\
0 & 1 \\
0 & 0 \\
\end{array} \right] \quad \text{constant} \\
\end{align*}
\]

\[
\begin{align*}
(x_1, x_2) = & \left[ \begin{array}{c|c}
1 & 0 \\
1 & 1 \\
0 & 0 \\
\end{array} \right] \quad \text{balanced} \\
\end{align*}
\]

\[
\begin{align*}
\text{if } f(x) = 0 & \quad \left[ \begin{array}{c|c}
0 & 0 \\
0 & 1 \\
0 & 0 \\
\end{array} \right] \\
\text{if } f(x) = 1 & \quad \left[ \begin{array}{c|c}
1 & 1 \\
1 & 1 \\
0 & 0 \\
\end{array} \right]
\end{align*}
\]

\[
\begin{align*}
\text{if } f(x) = 0 & \quad \left[ \begin{array}{c|c}
m_x & 0 \\
0 & m_x \\
\end{array} \right] \\
\text{if } f(x) = 1 & \quad \left[ \begin{array}{c|c}
m_x & m_x \\
0 & m_x \\
\end{array} \right]
\end{align*}
\]

\[
\begin{align*}
\text{if } f(x) = 0 & \quad \left[ \begin{array}{c|c}
\frac{1}{\sqrt{2}} & \left( \begin{array}{c}
0 \\
1
\end{array} \right) \\
0 & \left( \begin{array}{c}
0 \\
1
\end{array} \right)
\end{array} \right] \\
\text{if } f(x) = 1 & \quad \left[ \begin{array}{c|c}
\frac{1}{\sqrt{2}} & \left( \begin{array}{c}
1 \\
0
\end{array} \right) \\
-\frac{1}{\sqrt{2}} & \left( \begin{array}{c}
1 \\
0
\end{array} \right)
\end{array} \right]
\end{align*}
\]
These two results can be regrouped into two expressions:

\[ |x> = \frac{1}{\sqrt{2}} \begin{pmatrix} 102 - 117 \end{pmatrix} \Rightarrow |y> \Rightarrow (-1)^f(x) |x> \begin{pmatrix} 102 - 117 \end{pmatrix} \]

The **Deutsch's algorithm**

[Diagram of Deutsch's algorithm]

\[ |140> = |102 \otimes 117 \]

\[ |141> = H |102 \otimes 117 = \left[ \frac{102 + 117}{\sqrt{2}} \right] \left[ \frac{102 - 117}{\sqrt{2}} \right] \]

\[ U_f |141> = (-1)^f(0) \frac{1}{\sqrt{2}} |102 - 117> + (-1)^f(1) \frac{1}{\sqrt{2}} |102 + 117> \]

If \( f(0) = f(1) \) then

\[ |141> = (-1)^f(0) \frac{1}{\sqrt{2}} |102 + 117> + (-1)^f(1) \frac{1}{\sqrt{2}} |102 - 117> \]

\[ |0> \text{ depending if } f(0) = f(1) = 0 \text{ or } 1 \]

If \( f(0) \neq f(1) \) then

\[ |141> = (-1)^f(0) \frac{1}{\sqrt{2}} |102 - 117> + (-1)^f(1) \frac{1}{\sqrt{2}} |102 + 117> \]

\[ 142> = U_f |141> = \begin{cases} \frac{102 + 117}{\sqrt{2}} \left( \frac{102 - 117}{\sqrt{2}} \right) & \text{if } f(0) = f(1) \\ \frac{102 - 117}{\sqrt{2}} \left( \frac{102 + 117}{\sqrt{2}} \right) & \text{if } f(0) \neq f(1) \end{cases} \]
Finally, the final Hadamard gate acting on the first qubit gives

\[ |y_3 \rangle = \begin{cases} 
  \pm 10 \left[ \frac{10 \pm 1 i}{\sqrt{2}} \right] & \text{if } f(0) = f(1) \\
  \pm 11 \left[ \frac{10 \pm 1 i}{\sqrt{2}} \right] & \text{if } f(0) \neq f(1)
\end{cases} \]

but \( f(0) \oplus f(1) = 0 \) if \( f(0) = f(1) \) see table top of pages 5.

So, we can rewrite \( |y_3 \rangle \) more concisely

\[ |y_3 \rangle = \pm |f(0) \oplus f(1) \rangle \left[ \frac{10 \pm 1 i}{\sqrt{2}} \right] \]

so we have accomplished our goal! By measuring the first qubit we may determine \( f(0) \oplus f(1) \). This is a global property of \( f(x) \)... using only one evaluation of \( f(x) \)! This is faster than with a classical apparatus, which would require at least 2 evaluations.
The Deutsch-Jozsa Algorithm

Alice in Amsterdam selects a number \( x \) between 0 and \( 2^{n-1} \) and sends it to Bob in Boston. Bob calculates \( f(x) \) and replies with the result 0 or 1.

If \( f(x) \) is either constant or balanced, it is:
- equal to 1 for exactly half \( x \), and 0 for the other half.
- Alice's goal is to determine if \( f(x) \) is constant or balanced by corresponding with Bob as little as possible.

Classically, it will take \( \frac{2^n}{2} + 1 \) communications with qubits, Alice can achieve this in ONE query!

Alice has a n-qubit register to store her query in.
Bob has one qubit.
Bob will evaluate \( f(x) \) using quantum parallelism.

The quantum circuit looks as follows:

```
1) |H^m|
   --------
2) U_f
   |
   |
   |
   |
3) |H^m|
   --------
```

Input:
- \( |0\rangle \), \( |1\rangle \), \( |y\rangle \), \( |y \oplus f(x)\rangle \), \( |14_0\rangle \), \( |14_1\rangle \), \( |14_2\rangle \), \( |14_3\rangle \), \( |14_4\rangle \), \( |14_5\rangle \), \( |14_6\rangle \), \( |14_7\rangle \), \( |14_8\rangle \), \( |14_9\rangle \), \( |14_{10}\rangle \), \( |14_{11}\rangle \), \( |14_{12}\rangle \), \( |14_{13}\rangle \), \( |14_{14}\rangle \), \( |14_{15}\rangle \)
Input state $\ket{40} = \ket{0} \otimes \ket{1}$

$$
\ket{41} = \sum_{x} \frac{\ket{x}}{\sqrt{2^m}} \left[ \frac{\ket{0} - \ket{1}}{\sqrt{2}} \right]
$$

query is a superposition of all states. 5 answer register is an evenly weighted superposition of 0 and 1.

$s(x)$ is affecting the amplitude of each term in the qubit register created by Alice.

4.3. For a single qubit

$$
H |x\rangle = \sum_{z=0,1} (-1)^x z \cdot \frac{|z\rangle}{\sqrt{2}}
$$

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

$|x\rangle = |0\rangle \Rightarrow H |0\rangle = \frac{1}{\sqrt{2}} \left( |0\rangle + |1\rangle \right) = \frac{(-1)^0 \cdot 0}{\sqrt{2}} |0\rangle + \frac{(-1)^1 \cdot 1}{\sqrt{2}} |1\rangle$

$|x\rangle = |1\rangle \Rightarrow H |1\rangle = \frac{1}{\sqrt{2}} \left( |0\rangle - |1\rangle \right) = \frac{(-1)^0 \cdot 0}{\sqrt{2}} |0\rangle + \frac{(-1)^1 \cdot 1}{\sqrt{2}} |1\rangle$

So for the Hadamard transform, we have

$$
H^\otimes m \ket{z_1, \ldots, z_m} = \sum_{z_1, \ldots, z_m} (-1)^{z_1 \cdot z_1 + \cdots + z_m \cdot z_m} \frac{|z_1, \ldots, z_m\rangle}{\sqrt{2^m}}
$$
$$H^{\otimes n} |x\rangle = \sum_{z} \frac{(-1)^{x \cdot z}}{\sqrt{2^n}} |z\rangle$$

$x \cdot z$ = bit-wise inner product of $x$ and $z$ (modulo 2).

$$\Rightarrow |\psi_{2}\rangle = \sum_{z} \frac{(-1)^{x \cdot z} + f(x)}{\sqrt{2^n}} |z\rangle \left[ \frac{[10^2 - 117]}{\sqrt{2}} \right]$$

Alice now derives the query register.

The amplitude for the state $|0\rangle^{\otimes m}$ is

$$\frac{2}{\sqrt{2^n}} \sum_{x} f(x)$$

If $f(x)$ = constant, amplitude is $(-1)^{0}$ or $(-1)^{1} < 2^n$ times.

But $|\psi_{2}\rangle$ has unit length $\Rightarrow$ all other amplitudes must be 0.

$\Rightarrow$ an observation will lead 0$^2$ for all qubits in the query register.

If $f$ is balanced, we have an even number of terms $\Rightarrow$ half will be positive, half negative, and their norms will exactly balance out. $\Rightarrow$ there is a zero amplitude for the state $|0\rangle^{\otimes m}$. So, a measurement must leave a result other than 0 on at least one qubit in the query register.
Summarizing,

If the measure all as then the function is constant; otherwise the function is balanced.

Deutsch's problem is not unfortunately an important problem.

There are algorithms which are quantum versions of the Fourier transform – Solov'ev's algorithm for factoring and discrete algorithm. Also, the Grover or quantum search algorithm.