Today's outline - February 17, 2022

Today's outline - February 17, 2022

- Deutch-Josza problem

Today's outline - February 17, 2022

- Deutch-Josza problem
- Bernstein-Vazirani problem

Today's outline - February 17, 2022

- Deutch-Josza problem
- Bernstein-Vazirani problem
- Mermin's interpretation of parallelism

Today's outline - February 17, 2022

- Deutch-Josza problem
- Bernstein-Vazirani problem
- Mermin's interpretation of parallelism
- Simon's problem

Today's outline - February 17, 2022

- Deutch-Josza problem
- Bernstein-Vazirani problem
- Mermin's interpretation of parallelism
- Simon's problem

Reading Assignment: Chapter 7.7-7.8

Today's outline - February 17, 2022

- Deutch-Josza problem
- Bernstein-Vazirani problem
- Mermin's interpretation of parallelism
- Simon's problem

Reading Assignment: Chapter 7.7-7.8

Homework Assignment #05:
Chapter 7:1,3,4
due Thursday, February 24, 2022

The Deutsch-Jozsa problem

This is a multi-qubit generalization of the Deutsch problem where a function is balanced if an equal number of input values return 0 and 1

The Deutsch-Jozsa problem

This is a multi-qubit generalization of the Deutsch problem where a function is balanced if an equal number of input values return 0 and 1

Given a function $f : \mathbf{Z}_{2^n} \mapsto \mathbf{Z}_2$ that is known to be either constant or balanced, and a quantum oracle $U_f : |x\rangle|y\rangle \rightarrow |x\rangle|y \oplus f(x)\rangle$, determine whether the function f is constant or balanced

The Deutsch-Jozsa problem

This is a multi-qubit generalization of the Deutsch problem where a function is balanced if an equal number of input values return 0 and 1

Given a function $f : \mathbf{Z}_{2^n} \mapsto \mathbf{Z}_2$ that is known to be either constant or balanced, and a quantum oracle $U_f : |x\rangle|y\rangle \to |x\rangle|y \oplus f(x)\rangle$, determine whether the function f is constant or balanced

Start by using the $\phi = \pi$ phase change subroutine to negate terms of the superposition of basis vectors $|x\rangle$ with $f(x) = 1$ which returns

The Deutsch-Jozsa problem

This is a multi-qubit generalization of the Deutsch problem where a function is balanced if an equal number of input values return 0 and 1

Given a function $f : \mathbf{Z}_{2^n} \mapsto \mathbf{Z}_2$ that is known to be either constant or balanced, and a quantum oracle $U_f : |x\rangle|y\rangle \rightarrow |x\rangle|y \oplus f(x)\rangle$, determine whether the function f is constant or balanced

Start by using the $\phi = \pi$ phase change subroutine to negate terms of the superposition of basis vectors $|x\rangle$ with $f(x) = 1$ which returns

$$|\psi\rangle = \frac{1}{\sqrt{N}} \sum_{i=0}^{N-1} (-1)^{f(i)} |i\rangle$$

The Deutsch-Jozsa problem

This is a multi-qubit generalization of the Deutsch problem where a function is balanced if an equal number of input values return 0 and 1

Given a function $f : \mathbf{Z}_{2^n} \mapsto \mathbf{Z}_2$ that is known to be either constant or balanced, and a quantum oracle $U_f : |x\rangle|y\rangle \rightarrow |x\rangle|y \oplus f(x)\rangle$, determine whether the function f is constant or balanced

Start by using the $\phi = \pi$ phase change subroutine to negate terms of the superposition of basis vectors $|x\rangle$ with $f(x) = 1$ which returns

$$|\psi\rangle = \frac{1}{\sqrt{N}} \sum_{i=0}^{N-1} (-1)^{f(i)} |i\rangle$$

Next apply the Walsh transform to $|\psi\rangle$ recalling that for a vector $|r\rangle$, the Walsh transform is

The Deutsch-Jozsa problem

This is a multi-qubit generalization of the Deutsch problem where a function is balanced if an equal number of input values return 0 and 1

Given a function $f : \mathbf{Z}_{2^n} \mapsto \mathbf{Z}_2$ that is known to be either constant or balanced, and a quantum oracle $U_f : |x\rangle|y\rangle \rightarrow |x\rangle|y \oplus f(x)\rangle$, determine whether the function f is constant or balanced

Start by using the $\phi = \pi$ phase change subroutine to negate terms of the superposition of basis vectors $|x\rangle$ with $f(x) = 1$ which returns

$$|\psi\rangle = \frac{1}{\sqrt{N}} \sum_{i=0}^{N-1} (-1)^{f(i)} |i\rangle$$

Next apply the Walsh transform to $|\psi\rangle$ recalling that for a vector $|r\rangle$, the Walsh transform is

$$W|r\rangle = \frac{1}{\sqrt{N}} \sum_{s=0}^{N-1} (-1)^{r \cdot s} |s\rangle$$

The Deutsch-Jozsa problem

This is a multi-qubit generalization of the Deutsch problem where a function is balanced if an equal number of input values return 0 and 1

Given a function $f : \mathbf{Z}_{2^n} \mapsto \mathbf{Z}_2$ that is known to be either constant or balanced, and a quantum oracle $U_f : |x\rangle|y\rangle \rightarrow |x\rangle|y \oplus f(x)\rangle$, determine whether the function f is constant or balanced

Start by using the $\phi = \pi$ phase change subroutine to negate terms of the superposition of basis vectors $|x\rangle$ with $f(x) = 1$ which returns

$$|\psi\rangle = \frac{1}{\sqrt{N}} \sum_{i=0}^{N-1} (-1)^{f(i)} |i\rangle$$

Next apply the Walsh transform to $|\psi\rangle$ recalling that for a vector $|r\rangle$, the Walsh transform is

$$W|r\rangle = \frac{1}{\sqrt{N}} \sum_{s=0}^{N-1} (-1)^{r \cdot s} |s\rangle$$

$$|\phi\rangle = W|\psi\rangle$$

The Deutsch-Jozsa problem

This is a multi-qubit generalization of the Deutsch problem where a function is balanced if an equal number of input values return 0 and 1

Given a function $f : \mathbf{Z}_{2^n} \mapsto \mathbf{Z}_2$ that is known to be either constant or balanced, and a quantum oracle $U_f : |x\rangle|y\rangle \rightarrow |x\rangle|y \oplus f(x)\rangle$, determine whether the function f is constant or balanced

Start by using the $\phi = \pi$ phase change subroutine to negate terms of the superposition of basis vectors $|x\rangle$ with $f(x) = 1$ which returns

$$|\psi\rangle = \frac{1}{\sqrt{N}} \sum_{i=0}^{N-1} (-1)^{f(i)} |i\rangle$$

Next apply the Walsh transform to $|\psi\rangle$ recalling that for a vector $|r\rangle$, the Walsh transform is

$$W|r\rangle = \frac{1}{\sqrt{N}} \sum_{s=0}^{N-1} (-1)^{r \cdot s} |s\rangle$$

$$|\phi\rangle = W|\psi\rangle = \frac{1}{N} \sum_{i=0}^{N-1} \left((-1)^{f(i)} \sum_{j=0}^{N-1} (-1)^{i \cdot j} |j\rangle \right)$$

The Deutsch-Jozsa problem

This is a multi-qubit generalization of the Deutsch problem where a function is balanced if an equal number of input values return 0 and 1

Given a function $f : \mathbf{Z}_{2^n} \mapsto \mathbf{Z}_2$ that is known to be either constant or balanced, and a quantum oracle $U_f : |x\rangle|y\rangle \rightarrow |x\rangle|y \oplus f(x)\rangle$, determine whether the function f is constant or balanced

Start by using the $\phi = \pi$ phase change subroutine to negate terms of the superposition of basis vectors $|x\rangle$ with $f(x) = 1$ which returns

$$|\psi\rangle = \frac{1}{\sqrt{N}} \sum_{i=0}^{N-1} (-1)^{f(i)} |i\rangle$$

Next apply the Walsh transform to $|\psi\rangle$ recalling that for a vector $|r\rangle$, the Walsh transform is

$$W|r\rangle = \frac{1}{\sqrt{N}} \sum_{s=0}^{N-1} (-1)^{r \cdot s} |s\rangle$$

$$|\phi\rangle = W|\psi\rangle = \frac{1}{N} \sum_{i=0}^{N-1} \left((-1)^{f(i)} \sum_{j=0}^{N-1} (-1)^{i \cdot j} |j\rangle \right)$$

For each vector $|i\rangle$ in the sum that makes up $|\psi\rangle$, the Walsh transform applies a sign change depending on the number of common 1 bits between $|i\rangle$ and $|j\rangle$

The Deutsch-Jozsa problem

$$|\phi\rangle = \frac{1}{N}\sum_{i=0}^{N-1}\left((-1)^{f(i)}\sum_{j=0}^{N-1}(-1)^{i\cdot j}|j\rangle\right)$$

The Deutsch-Jozsa problem

$$|\phi\rangle = \frac{1}{N}\sum_{i=0}^{N-1}\left((-1)^{f(i)}\sum_{j=0}^{N-1}(-1)^{i\cdot j}|j\rangle\right)$$

For constant f, $(-1)^{f(i)} = (-1)^{f(0)}$ is a global phase and can be pulled out of the sum

The Deutsch-Jozsa problem

$$|\phi\rangle = \frac{1}{N}\sum_{i=0}^{N-1}\left((-1)^{f(i)}\sum_{j=0}^{N-1}(-1)^{i\cdot j}|j\rangle\right)$$

For constant f, $(-1)^{f(i)} = (-1)^{f(0)}$ is a global phase and can be pulled out of the sum

$$|\phi\rangle = (-1)^{f(0)}\frac{1}{N}\sum_{j=0}^{N-1}\left(\sum_{i=0}^{N-1}(-1)^{i\cdot j}\right)|j\rangle$$

The Deutsch-Jozsa problem

$$|\phi\rangle = \frac{1}{N}\sum_{i=0}^{N-1}\left((-1)^{f(i)}\sum_{j=0}^{N-1}(-1)^{i\cdot j}|j\rangle\right)$$

For constant f, $(-1)^{f(i)} = (-1)^{f(0)}$ is a global phase and can be pulled out of the sum

$$|\phi\rangle = (-1)^{f(0)}\frac{1}{N}\sum_{j=0}^{N-1}\left(\sum_{i=0}^{N-1}(-1)^{i\cdot j}\right)|j\rangle$$

But

$$\sum_{x=0}^{N-1}(-1)^{x\cdot y} = \begin{cases} N & y=0 \\ 0 & y\neq 0\end{cases}$$

The Deutsch-Jozsa problem

$$|\phi\rangle = \frac{1}{N}\sum_{i=0}^{N-1}\left((-1)^{f(i)}\sum_{j=0}^{N-1}(-1)^{i\cdot j}|j\rangle\right)$$

For constant f, $(-1)^{f(i)} = (-1)^{f(0)}$ is a global phase and can be pulled out of the sum

$$|\phi\rangle = (-1)^{f(0)}\frac{1}{N}\sum_{j=0}^{N-1}\left(\sum_{i=0}^{N-1}(-1)^{i\cdot j}\right)|j\rangle$$

But

$$\sum_{x=0}^{N-1}(-1)^{x\cdot y} = \begin{cases} N & y=0 \\ 0 & y\neq 0 \end{cases}$$

$$= (-1)^{f(0)}\frac{1}{N}\sum_{i=0}^{N-1}(-1)^{i\cdot 0}|0\rangle$$

The Deutsch-Jozsa problem

$$|\phi\rangle = \frac{1}{N}\sum_{i=0}^{N-1}\left((-1)^{f(i)}\sum_{j=0}^{N-1}(-1)^{i\cdot j}|j\rangle\right)$$

For constant f, $(-1)^{f(i)} = (-1)^{f(0)}$ is a global phase and can be pulled out of the sum

But $$\sum_{x=0}^{N-1}(-1)^{x\cdot y} = \begin{cases} N & y = 0 \\ 0 & y \neq 0 \end{cases}$$

$$|\phi\rangle = (-1)^{f(0)}\frac{1}{N}\sum_{j=0}^{N-1}\left(\sum_{i=0}^{N-1}(-1)^{i\cdot j}\right)|j\rangle$$

$$= (-1)^{f(0)}\frac{1}{N}\sum_{i=0}^{N-1}(-1)^{i\cdot 0}|0\rangle = (-1)^{f(0)}|0\rangle$$

The Deutsch-Jozsa problem

$$|\phi\rangle = \frac{1}{N}\sum_{i=0}^{N-1}\left((-1)^{f(i)}\sum_{j=0}^{N-1}(-1)^{i\cdot j}|j\rangle\right)$$

For constant f, $(-1)^{f(i)} = (-1)^{f(0)}$ is a global phase and can be pulled out of the sum

$$|\phi\rangle = (-1)^{f(0)}\frac{1}{N}\sum_{j=0}^{N-1}\left(\sum_{i=0}^{N-1}(-1)^{i\cdot j}\right)|j\rangle$$

$$= (-1)^{f(0)}\frac{1}{N}\sum_{i=0}^{N-1}(-1)^{i\cdot 0}|0\rangle = (-1)^{f(0)}|0\rangle$$

But $$\sum_{x=0}^{N-1}(-1)^{x\cdot y} = \begin{cases} N & y=0 \\ 0 & y\neq 0 \end{cases}$$

For balanced f, $f(i) = 0$ when $i \in X_0$

The Deutsch-Jozsa problem

$$|\phi\rangle = \frac{1}{N}\sum_{i=0}^{N-1}\left((-1)^{f(i)}\sum_{j=0}^{N-1}(-1)^{i\cdot j}|j\rangle\right)$$

For constant f, $(-1)^{f(i)} = (-1)^{f(0)}$ is a global phase and can be pulled out of the sum

$$|\phi\rangle = (-1)^{f(0)}\frac{1}{N}\sum_{j=0}^{N-1}\left(\sum_{i=0}^{N-1}(-1)^{i\cdot j}\right)|j\rangle$$

$$= (-1)^{f(0)}\frac{1}{N}\sum_{i=0}^{N-1}(-1)^{i\cdot 0}|0\rangle = (-1)^{f(0)}|0\rangle$$

But $$\sum_{x=0}^{N-1}(-1)^{x\cdot y} = \begin{cases} N & y=0 \\ 0 & y\neq 0\end{cases}$$

For balanced f, $f(i) = 0$ when $i \in X_0$

$$|\phi\rangle = \frac{1}{N}\sum_{j=0}^{N-1}\left(\sum_{i\in X_0}(-1)^{i\cdot j} - \sum_{i\notin X_0}(-1)^{i\cdot j}\right)|j\rangle$$

The Deutsch-Jozsa problem

$$|\phi\rangle = \frac{1}{N}\sum_{i=0}^{N-1}\left((-1)^{f(i)}\sum_{j=0}^{N-1}(-1)^{i\cdot j}|j\rangle\right)$$

For constant f, $(-1)^{f(i)} = (-1)^{f(0)}$ is a global phase and can be pulled out of the sum

$$|\phi\rangle = (-1)^{f(0)}\frac{1}{N}\sum_{j=0}^{N-1}\left(\sum_{i=0}^{N-1}(-1)^{i\cdot j}\right)|j\rangle$$

$$= (-1)^{f(0)}\frac{1}{N}\sum_{i=0}^{N-1}(-1)^{i\cdot 0}|0\rangle = (-1)^{f(0)}|0\rangle$$

But $$\sum_{x=0}^{N-1}(-1)^{x\cdot y} = \begin{cases} N & y=0 \\ 0 & y\neq 0\end{cases}$$

For balanced f, $f(i) = 0$ when $i \in X_0$ and the two internal sums must cancel when $|j\rangle = |0\rangle$ but not otherwise

$$|\phi\rangle = \frac{1}{N}\sum_{j=0}^{N-1}\left(\sum_{i\in X_0}(-1)^{i\cdot j} - \sum_{i\notin X_0}(-1)^{i\cdot j}\right)|j\rangle$$

The Deutsch-Jozsa problem

$$|\phi\rangle = \frac{1}{N}\sum_{i=0}^{N-1}\left((-1)^{f(i)}\sum_{j=0}^{N-1}(-1)^{i\cdot j}|j\rangle\right)$$

For constant f, $(-1)^{f(i)} = (-1)^{f(0)}$ is a global phase and can be pulled out of the sum

But $$\sum_{x=0}^{N-1}(-1)^{x\cdot y} = \begin{cases} N & y=0 \\ 0 & y \neq 0 \end{cases}$$

For balanced f, $f(i) = 0$ when $i \in X_0$ and the two internal sums must cancel when $|j\rangle = |0\rangle$ but not otherwise

For constant $f(x)$, $|\phi\rangle = |0\rangle$

$$|\phi\rangle = (-1)^{f(0)}\frac{1}{N}\sum_{j=0}^{N-1}\left(\sum_{i=0}^{N-1}(-1)^{i\cdot j}\right)|j\rangle$$

$$= (-1)^{f(0)}\frac{1}{N}\sum_{i=0}^{N-1}(-1)^{i\cdot 0}|0\rangle = (-1)^{f(0)}|0\rangle$$

$$|\phi\rangle = \frac{1}{N}\sum_{j=0}^{N-1}\left(\sum_{i\in X_0}(-1)^{i\cdot j} - \sum_{i\notin X_0}(-1)^{i\cdot j}\right)|j\rangle$$

The Deutsch-Jozsa problem

$$|\phi\rangle = \frac{1}{N}\sum_{i=0}^{N-1}\left((-1)^{f(i)}\sum_{j=0}^{N-1}(-1)^{i\cdot j}|j\rangle\right)$$

For constant f, $(-1)^{f(i)} = (-1)^{f(0)}$ is a global phase and can be pulled out of the sum

But
$$\sum_{x=0}^{N-1}(-1)^{x\cdot y} = \begin{cases} N & y=0 \\ 0 & y\neq 0 \end{cases}$$

For balanced f, $f(i) = 0$ when $i \in X_0$ and the two internal sums must cancel when $|j\rangle = |0\rangle$ but not otherwise

For constant $f(x)$, $|\phi\rangle = |0\rangle$

For balanced $f(x)$, $|\phi\rangle = |j\rangle \neq |0\rangle$

$$|\phi\rangle = (-1)^{f(0)}\frac{1}{N}\sum_{j=0}^{N-1}\left(\sum_{i=0}^{N-1}(-1)^{i\cdot j}\right)|j\rangle$$

$$= (-1)^{f(0)}\frac{1}{N}\sum_{i=0}^{N-1}(-1)^{i\cdot 0}|0\rangle = (-1)^{f(0)}|0\rangle$$

$$|\phi\rangle = \frac{1}{N}\sum_{j=0}^{N-1}\left(\sum_{i\in X_0}(-1)^{i\cdot j} - \sum_{i\notin X_0}(-1)^{i\cdot j}\right)|j\rangle$$

The Deutsch-Jozsa problem

$$|\phi\rangle = \frac{1}{N}\sum_{i=0}^{N-1}\left((-1)^{f(i)}\sum_{j=0}^{N-1}(-1)^{i\cdot j}|j\rangle\right)$$

For constant $f(x)$, $|\phi\rangle = |0\rangle$

For balanced $f(x)$, $|\phi\rangle = |j\rangle \neq |0\rangle$

For constant f, $(-1)^{f(i)} = (-1)^{f(0)}$ is a global phase and can be pulled out of the sum

$$|\phi\rangle = (-1)^{f(0)}\frac{1}{N}\sum_{j=0}^{N-1}\left(\sum_{i=0}^{N-1}(-1)^{i\cdot j}\right)|j\rangle$$

But $$\sum_{x=0}^{N-1}(-1)^{x\cdot y} = \begin{cases} N & y=0 \\ 0 & y\neq 0 \end{cases}$$

$$= (-1)^{f(0)}\frac{1}{N}\sum_{i=0}^{N-1}(-1)^{i\cdot 0}|0\rangle = (-1)^{f(0)}|0\rangle$$

For balanced f, $f(i) = 0$ when $i \in X_0$ and the two internal sums must cancel when $|j\rangle = |0\rangle$ but not otherwise

$$|\phi\rangle = \frac{1}{N}\sum_{j=0}^{N-1}\left(\sum_{i\in X_0}(-1)^{i\cdot j} - \sum_{i\notin X_0}(-1)^{i\cdot j}\right)|j\rangle$$

This solves the Deutsch-Jozsa problem with a single call to U_f which is exponentially better than the classical solution

The Bernstein-Vazirani problem

The Bernstein-Vazirani problem is to determine the value of an unknown string u of bit length n using only queries of the form $q \cdot u$

The Bernstein-Vazirani problem

The Bernstein-Vazirani problem is to determine the value of an unknown string u of bit length n using only queries of the form $q \cdot u$

The quantum algorithm can solve this using a single query to a transformation U_{f_u} where $f_u(q) = q \cdot u \mod 2$ and

The Bernstein-Vazirani problem

The Bernstein-Vazirani problem is to determine the value of an unknown string u of bit length n using only queries of the form $q \cdot u$

The quantum algorithm can solve this using a single query to a transformation U_{f_u} where $f_u(q) = q \cdot u \mod 2$ and

$$U_{f_u} : |q\rangle|b\rangle \mapsto |q\rangle|b \oplus f_u(q)\rangle$$

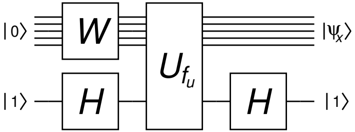
The Bernstein-Vazirani problem

The Bernstein-Vazirani problem is to determine the value of an unknown string u of bit length n using only queries of the form $q \cdot u$

The quantum algorithm can solve this using a single query to a transformation U_{f_u} where $f_u(q) = q \cdot u \mod 2$ and

$$U_{f_u} : |q\rangle|b\rangle \mapsto |q\rangle|b \oplus f_u(q)\rangle$$

This is solved by starting with the circuit that was used to apply the $\phi = \pi$ phase change which gives



The Bernstein-Vazirani problem

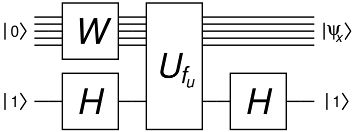
The Bernstein-Vazirani problem is to determine the value of an unknown string u of bit length n using only queries of the form $q \cdot u$

The quantum algorithm can solve this using a single query to a transformation U_{f_u} where $f_u(q) = q \cdot u \mod 2$ and

$$U_{f_u} : |q\rangle|b\rangle \mapsto |q\rangle|b \oplus f_u(q)\rangle$$

This is solved by starting with the circuit that was used to apply the $\phi = \pi$ phase change which gives

$$|\psi_X\rangle = \frac{1}{\sqrt{N}} \sum_{q=0}^{N-1} (-1)^{f_u(q)} |q\rangle$$



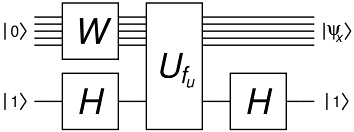
The Bernstein-Vazirani problem

The Bernstein-Vazirani problem is to determine the value of an unknown string u of bit length n using only queries of the form $q \cdot u$

The quantum algorithm can solve this using a single query to a transformation U_{f_u} where $f_u(q) = q \cdot u \mod 2$ and

$$U_{f_u} : |q\rangle|b\rangle \mapsto |q\rangle|b \oplus f_u(q)\rangle$$

This is solved by starting with the circuit that was used to apply the $\phi = \pi$ phase change which gives



$$|\psi_X\rangle = \frac{1}{\sqrt{N}} \sum_{q=0}^{N-1} (-1)^{f_u(q)} |q\rangle = \frac{1}{\sqrt{N}} \sum_{q=0}^{N-1} (-1)^{u \cdot q} |q\rangle$$

If the Walsh-Hadamard transformation is now applied to $|\psi_X\rangle$ we have

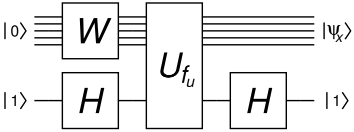
The Bernstein-Vazirani problem

The Bernstein-Vazirani problem is to determine the value of an unknown string u of bit length n using only queries of the form $q \cdot u$

The quantum algorithm can solve this using a single query to a transformation U_{f_u} where $f_u(q) = q \cdot u \mod 2$ and

$$U_{f_u} : |q\rangle|b\rangle \mapsto |q\rangle|b \oplus f_u(q)\rangle$$

This is solved by starting with the circuit that was used to apply the $\phi = \pi$ phase change which gives



$$|\psi_X\rangle = \frac{1}{\sqrt{N}} \sum_{q=0}^{N-1} (-1)^{f_u(q)} |q\rangle = \frac{1}{\sqrt{N}} \sum_{q=0}^{N-1} (-1)^{u \cdot q} |q\rangle$$

If the Walsh-Hadamard transformation is now applied to $|\psi_X\rangle$ we have

$$W|\psi_X\rangle = W \left(\frac{1}{\sqrt{N}} \sum_{q=0}^{N-1} (-1)^{u \cdot q} |q\rangle \right)$$

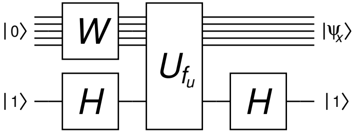
The Bernstein-Vazirani problem

The Bernstein-Vazirani problem is to determine the value of an unknown string u of bit length n using only queries of the form $q \cdot u$

The quantum algorithm can solve this using a single query to a transformation U_{f_u} where $f_u(q) = q \cdot u \mod 2$ and

$$U_{f_u} : |q\rangle|b\rangle \mapsto |q\rangle|b \oplus f_u(q)\rangle$$

This is solved by starting with the circuit that was used to apply the $\phi = \pi$ phase change which gives



$$|\psi_X\rangle = \frac{1}{\sqrt{N}} \sum_{q=0}^{N-1} (-1)^{f_u(q)} |q\rangle = \frac{1}{\sqrt{N}} \sum_{q=0}^{N-1} (-1)^{u \cdot q} |q\rangle$$

If the Walsh-Hadamard transformation is now applied to $|\psi_X\rangle$ we have

$$W|\psi_X\rangle = W \left(\frac{1}{\sqrt{N}} \sum_{q=0}^{N-1} (-1)^{u \cdot q} |q\rangle \right) = \frac{1}{\sqrt{N}} \sum_{q=0}^{N-1} (-1)^{u \cdot q} W|q\rangle$$

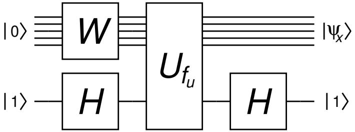
The Bernstein-Vazirani problem

The Bernstein-Vazirani problem is to determine the value of an unknown string u of bit length n using only queries of the form $q \cdot u$

The quantum algorithm can solve this using a single query to a transformation U_{f_u} where $f_u(q) = q \cdot u \mod 2$ and

$$U_{f_u} : |q\rangle|b\rangle \mapsto |q\rangle|b \oplus f_u(q)\rangle$$

This is solved by starting with the circuit that was used to apply the $\phi = \pi$ phase change which gives



$$|\psi_X\rangle = \frac{1}{\sqrt{N}} \sum_{q=0}^{N-1} (-1)^{f_u(q)} |q\rangle = \frac{1}{\sqrt{N}} \sum_{q=0}^{N-1} (-1)^{u \cdot q} |q\rangle$$

If the Walsh-Hadamard transformation is now applied to $|\psi_X\rangle$ we have

$$W|\psi_X\rangle = W\left(\frac{1}{\sqrt{N}} \sum_{q=0}^{N-1} (-1)^{u \cdot q} |q\rangle\right) = \frac{1}{\sqrt{N}} \sum_{q=0}^{N-1} (-1)^{u \cdot q} W|q\rangle = \frac{1}{N} \sum_{q=0}^{N-1} (-1)^{u \cdot q} \left(\sum_{z=0}^{N-1} (-1)^{q \cdot z} |z\rangle\right)$$

The Bernstein-Vazirani problem

$$W|\psi_X\rangle = \frac{1}{N}\sum_{q=0}^{N-1}(-1)^{u\cdot q}\left(\sum_{z=0}^{N-1}(-1)^{q\cdot z}|z\rangle\right)$$

The Bernstein-Vazirani problem

$$W|\psi_X\rangle = \frac{1}{N}\sum_{q=0}^{N-1}(-1)^{u\cdot q}\left(\sum_{z=0}^{N-1}(-1)^{q\cdot z}|z\rangle\right)$$

But from the discussion of the Hanning distance, we have that

$$(-1)^{u\cdot q+q\cdot z} \equiv (-1)^{(u\oplus z)\cdot q}$$

The Bernstein-Vazirani problem

$$W|\psi_x\rangle = \frac{1}{N}\sum_{q=0}^{N-1}(-1)^{u\cdot q}\left(\sum_{z=0}^{N-1}(-1)^{q\cdot z}|z\rangle\right)$$

$$= \frac{1}{N}\sum_{z=0}^{N-1}\left(\sum_{q=0}^{N-1}(-1)^{(u\oplus z)\cdot q}|z\rangle\right)$$

But from the discussion of the Hanning distance, we have that

$$(-1)^{u\cdot q+q\cdot z} \equiv (-1)^{(u\oplus z)\cdot q}$$

The Bernstein-Vazirani problem

$$W|\psi_x\rangle = \frac{1}{N}\sum_{q=0}^{N-1}(-1)^{u\cdot q}\left(\sum_{z=0}^{N-1}(-1)^{q\cdot z}|z\rangle\right)$$

$$= \frac{1}{N}\sum_{z=0}^{N-1}\left(\sum_{q=0}^{N-1}(-1)^{(u\oplus z)\cdot q}|z\rangle\right)$$

But from the discussion of the Hanning distance, we have that

$$(-1)^{u\cdot q+q\cdot z} \equiv (-1)^{(u\oplus z)\cdot q}$$

And the internal sum is zero unless $u\oplus z \equiv 0$ so only the term where $z \equiv u$ remains

The Bernstein-Vazirani problem

$$W|\psi_X\rangle = \frac{1}{N}\sum_{q=0}^{N-1}(-1)^{u\cdot q}\left(\sum_{z=0}^{N-1}(-1)^{q\cdot z}|z\rangle\right)$$

$$= \frac{1}{N}\sum_{z=0}^{N-1}\left(\sum_{q=0}^{N-1}(-1)^{(u\oplus z)\cdot q}|z\rangle\right)$$

$$= \frac{1}{N}\sum_{q=0}^{N-1}(-1)^{q\cdot 0}|u\rangle$$

But from the discussion of the Hanning distance, we have that

$$(-1)^{u\cdot q+q\cdot z} \equiv (-1)^{(u\oplus z)\cdot q}$$

And the internal sum is zero unless $u \oplus z \equiv 0$ so only the term where $z \equiv u$ remains

The Bernstein-Vazirani problem

$$W|\psi_X\rangle = \frac{1}{N}\sum_{q=0}^{N-1}(-1)^{u\cdot q}\left(\sum_{z=0}^{N-1}(-1)^{q\cdot z}|z\rangle\right)$$

$$= \frac{1}{N}\sum_{z=0}^{N-1}\left(\sum_{q=0}^{N-1}(-1)^{(u\oplus z)\cdot q}|z\rangle\right)$$

$$= \frac{1}{N}\sum_{q=0}^{N-1}(-1)^{q\cdot 0}|u\rangle = \frac{1}{N}N|u\rangle = |u\rangle$$

But from the discussion of the Hanning distance, we have that

$$(-1)^{u\cdot q+q\cdot z} \equiv (-1)^{(u\oplus z)\cdot q}$$

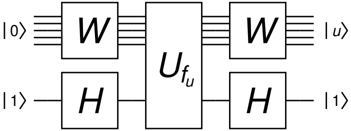
And the internal sum is zero unless $u\oplus z \equiv 0$ so only the term where $z \equiv u$ remains

The Bernstein-Vazirani problem

$$W|\psi_X\rangle = \frac{1}{N}\sum_{q=0}^{N-1}(-1)^{u\cdot q}\left(\sum_{z=0}^{N-1}(-1)^{q\cdot z}|z\rangle\right)$$

$$= \frac{1}{N}\sum_{z=0}^{N-1}\left(\sum_{q=0}^{N-1}(-1)^{(u\oplus z)\cdot q}|z\rangle\right)$$

$$= \frac{1}{N}\sum_{q=0}^{N-1}(-1)^{q\cdot 0}|u\rangle = \frac{1}{N}N|u\rangle = |u\rangle$$



But from the discussion of the Hanning distance, we have that

$$(-1)^{u\cdot q+q\cdot z} \equiv (-1)^{(u\oplus z)\cdot q}$$

And the internal sum is zero unless $u \oplus z \equiv 0$ so only the term where $z \equiv u$ remains

The Bernstein-Vazirani problem

$$W|\psi_X\rangle = \frac{1}{N}\sum_{q=0}^{N-1}(-1)^{u\cdot q}\left(\sum_{z=0}^{N-1}(-1)^{q\cdot z}|z\rangle\right)$$

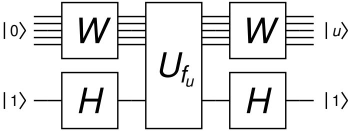
$$= \frac{1}{N}\sum_{z=0}^{N-1}\left(\sum_{q=0}^{N-1}(-1)^{(u\oplus z)\cdot q}|z\rangle\right)$$

$$= \frac{1}{N}\sum_{q=0}^{N-1}(-1)^{q\cdot 0}|u\rangle = \frac{1}{N}N|u\rangle = |u\rangle$$

But from the discussion of the Hanning distance, we have that

$$(-1)^{u\cdot q+q\cdot z} \equiv (-1)^{(u\oplus z)\cdot q}$$

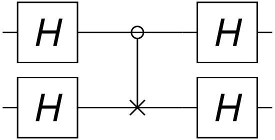
And the internal sum is zero unless $u\oplus z \equiv 0$ so only the term where $z \equiv u$ remains



This illustrates a common interpretation of how quantum circuits work, that is using parallelism to perform a computation on all possible inputs then manipulate the resulting superposition to get the result

Mermin's interpretation

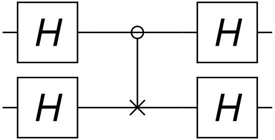
David Mermin proposed a simpler interpretation for how quantum algorithms and the solution to the Bernstein-Vazirani problem, in particular



Mermin's interpretation

David Mermin proposed a simpler interpretation for how quantum algorithms and the solution to the Bernstein-Vazirani problem, in particular

Consider a C_{not} acting on the Hadamard basis

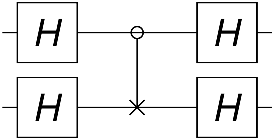


Mermin's interpretation

David Mermin proposed a simpler interpretation for how quantum algorithms and the solution to the Bernstein-Vazirani problem, in particular

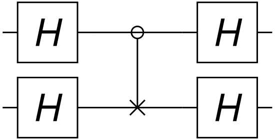
Consider a C_{not} acting on the Hadamard basis

$$C_{not}|++\rangle = C_{not}\tfrac{1}{2}(|00\rangle + |01\rangle + |10\rangle + |11\rangle)$$



Mermin's interpretation

David Mermin proposed a simpler interpretation for how quantum algorithms and the solution to the Bernstein-Vazirani problem, in particular

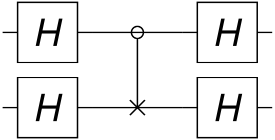


Consider a C_{not} acting on the Hadamard basis

$$C_{not}|++\rangle = C_{not}\tfrac{1}{2}(|00\rangle + |01\rangle + |10\rangle + |11\rangle) = \tfrac{1}{2}(|00\rangle + |01\rangle + |11\rangle + |10\rangle) = |++\rangle$$

Mermin's interpretation

David Mermin proposed a simpler interpretation for how quantum algorithms and the solution to the Bernstein-Vazirani problem, in particular



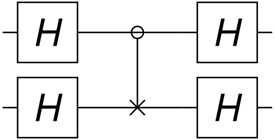
Consider a C_{not} acting on the Hadamard basis

$$C_{not}|++\rangle = C_{not}\tfrac{1}{2}(|00\rangle + |01\rangle + |10\rangle + |11\rangle) = \tfrac{1}{2}(|00\rangle + |01\rangle + |11\rangle + |10\rangle) = |++\rangle$$

$$C_{not}|+-\rangle = C_{not}\tfrac{1}{2}(|00\rangle - |01\rangle + |10\rangle - |11\rangle)$$

Mermin's interpretation

David Mermin proposed a simpler interpretation for how quantum algorithms and the solution to the Bernstein-Vazirani problem, in particular



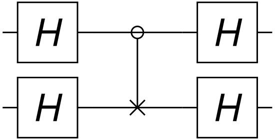
Consider a C_{not} acting on the Hadamard basis

$$C_{not}|++\rangle = C_{not}\tfrac{1}{2}(|00\rangle + |01\rangle + |10\rangle + |11\rangle) = \tfrac{1}{2}(|00\rangle + |01\rangle + |11\rangle + |10\rangle) = |++\rangle$$

$$C_{not}|+-\rangle = C_{not}\tfrac{1}{2}(|00\rangle - |01\rangle + |10\rangle - |11\rangle) = \tfrac{1}{2}(|00\rangle - |01\rangle + |11\rangle - |10\rangle) = |--\rangle$$

Mermin's interpretation

David Mermin proposed a simpler interpretation for how quantum algorithms and the solution to the Bernstein-Vazirani problem, in particular



Consider a C_{not} acting on the Hadamard basis

$$C_{not}|++\rangle = C_{not}\tfrac{1}{2}(|00\rangle + |01\rangle + |10\rangle + |11\rangle) = \tfrac{1}{2}(|00\rangle + |01\rangle + |11\rangle + |10\rangle) = |++\rangle$$

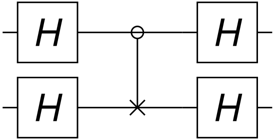
$$C_{not}|+-\rangle = C_{not}\tfrac{1}{2}(|00\rangle - |01\rangle + |10\rangle - |11\rangle) = \tfrac{1}{2}(|00\rangle - |01\rangle + |11\rangle - |10\rangle) = |--\rangle$$

$$C_{not}|-+\rangle = C_{not}\tfrac{1}{2}(|00\rangle + |01\rangle - |10\rangle - |11\rangle)$$

Mermin's interpretation

David Mermin proposed a simpler interpretation for how quantum algorithms and the solution to the Bernstein-Vazirani problem, in particular

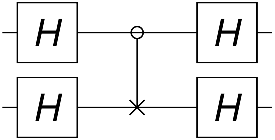
Consider a C_{not} acting on the Hadamard basis



$$C_{not}|++\rangle = C_{not}\tfrac{1}{2}(|00\rangle + |01\rangle + |10\rangle + |11\rangle) = \tfrac{1}{2}(|00\rangle + |01\rangle + |11\rangle + |10\rangle) = |++\rangle$$
$$C_{not}|+-\rangle = C_{not}\tfrac{1}{2}(|00\rangle - |01\rangle + |10\rangle - |11\rangle) = \tfrac{1}{2}(|00\rangle - |01\rangle + |11\rangle - |10\rangle) = |--\rangle$$
$$C_{not}|-+\rangle = C_{not}\tfrac{1}{2}(|00\rangle + |01\rangle - |10\rangle - |11\rangle) = \tfrac{1}{2}(|00\rangle + |01\rangle - |11\rangle - |10\rangle) = |-+\rangle$$

Mermin's interpretation

David Mermin proposed a simpler interpretation for how quantum algorithms and the solution to the Bernstein-Vazirani problem, in particular



Consider a C_{not} acting on the Hadamard basis

$$C_{not}|++\rangle = C_{not}\tfrac{1}{2}(|00\rangle + |01\rangle + |10\rangle + |11\rangle) = \tfrac{1}{2}(|00\rangle + |01\rangle + |11\rangle + |10\rangle) = |++\rangle$$

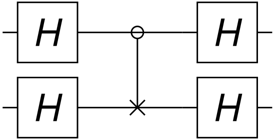
$$C_{not}|+-\rangle = C_{not}\tfrac{1}{2}(|00\rangle - |01\rangle + |10\rangle - |11\rangle) = \tfrac{1}{2}(|00\rangle - |01\rangle + |11\rangle - |10\rangle) = |--\rangle$$

$$C_{not}|-+\rangle = C_{not}\tfrac{1}{2}(|00\rangle + |01\rangle - |10\rangle - |11\rangle) = \tfrac{1}{2}(|00\rangle + |01\rangle - |11\rangle - |10\rangle) = |-+\rangle$$

$$C_{not}|--\rangle = C_{not}\tfrac{1}{2}(|00\rangle - |01\rangle - |10\rangle + |11\rangle)$$

Mermin's interpretation

David Mermin proposed a simpler interpretation for how quantum algorithms and the solution to the Bernstein-Vazirani problem, in particular

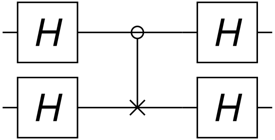


Consider a C_{not} acting on the Hadamard basis

$$C_{not}|++\rangle = C_{not}\tfrac{1}{2}(|00\rangle + |01\rangle + |10\rangle + |11\rangle) = \tfrac{1}{2}(|00\rangle + |01\rangle + |11\rangle + |10\rangle) = |++\rangle$$
$$C_{not}|+-\rangle = C_{not}\tfrac{1}{2}(|00\rangle - |01\rangle + |10\rangle - |11\rangle) = \tfrac{1}{2}(|00\rangle - |01\rangle + |11\rangle - |10\rangle) = |--\rangle$$
$$C_{not}|-+\rangle = C_{not}\tfrac{1}{2}(|00\rangle + |01\rangle - |10\rangle - |11\rangle) = \tfrac{1}{2}(|00\rangle + |01\rangle - |11\rangle - |10\rangle) = |-+\rangle$$
$$C_{not}|--\rangle = C_{not}\tfrac{1}{2}(|00\rangle - |01\rangle - |10\rangle + |11\rangle) = \tfrac{1}{2}(|00\rangle - |01\rangle - |11\rangle + |10\rangle) = |+-\rangle$$

Mermin's interpretation

David Mermin proposed a simpler interpretation for how quantum algorithms and the solution to the Bernstein-Vazirani problem, in particular



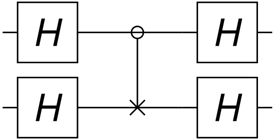
Consider a C_{not} acting on the Hadamard basis

$$C_{not}|++\rangle = C_{not}\tfrac{1}{2}(|00\rangle + |01\rangle + |10\rangle + |11\rangle) = \tfrac{1}{2}(|00\rangle + |01\rangle + |11\rangle + |10\rangle) = |++\rangle$$
$$C_{not}|+-\rangle = C_{not}\tfrac{1}{2}(|00\rangle - |01\rangle + |10\rangle - |11\rangle) = \tfrac{1}{2}(|00\rangle - |01\rangle + |11\rangle - |10\rangle) = |--\rangle$$
$$C_{not}|-+\rangle = C_{not}\tfrac{1}{2}(|00\rangle + |01\rangle - |10\rangle - |11\rangle) = \tfrac{1}{2}(|00\rangle + |01\rangle - |11\rangle - |10\rangle) = |-+\rangle$$
$$C_{not}|--\rangle = C_{not}\tfrac{1}{2}(|00\rangle - |01\rangle - |10\rangle + |11\rangle) = \tfrac{1}{2}(|00\rangle - |01\rangle - |11\rangle + |10\rangle) = |+-\rangle$$

If we then apply the Hadamard transform to each bit the resulting truth table becomes

Mermin's interpretation

David Mermin proposed a simpler interpretation for how quantum algorithms and the solution to the Bernstein-Vazirani problem, in particular



Consider a C_{not} acting on the Hadamard basis

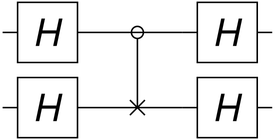
$$
\begin{aligned}
C_{not}|++\rangle &= C_{not}\tfrac{1}{2}(|00\rangle + |01\rangle + |10\rangle + |11\rangle) = \tfrac{1}{2}(|00\rangle + |01\rangle + |11\rangle + |10\rangle) = |++\rangle \\
C_{not}|+-\rangle &= C_{not}\tfrac{1}{2}(|00\rangle - |01\rangle + |10\rangle - |11\rangle) = \tfrac{1}{2}(|00\rangle - |01\rangle + |11\rangle - |10\rangle) = |--\rangle \\
C_{not}|-+\rangle &= C_{not}\tfrac{1}{2}(|00\rangle + |01\rangle - |10\rangle - |11\rangle) = \tfrac{1}{2}(|00\rangle + |01\rangle - |11\rangle - |10\rangle) = |-+\rangle \\
C_{not}|--\rangle &= C_{not}\tfrac{1}{2}(|00\rangle - |01\rangle - |10\rangle + |11\rangle) = \tfrac{1}{2}(|00\rangle - |01\rangle - |11\rangle + |10\rangle) = |+-\rangle
\end{aligned}
$$

If we then apply the Hadamard transform to each bit the resulting truth table becomes

Initial Final

Mermin's interpretation

David Mermin proposed a simpler interpretation for how quantum algorithms and the solution to the Bernstein-Vazirani problem, in particular



Consider a C_{not} acting on the Hadamard basis

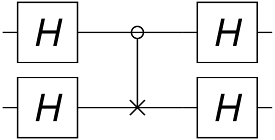
$$C_{not}|++\rangle = C_{not}\tfrac{1}{2}(|00\rangle + |01\rangle + |10\rangle + |11\rangle) = \tfrac{1}{2}(|00\rangle + |01\rangle + |11\rangle + |10\rangle) = |++\rangle$$
$$C_{not}|+-\rangle = C_{not}\tfrac{1}{2}(|00\rangle - |01\rangle + |10\rangle - |11\rangle) = \tfrac{1}{2}(|00\rangle - |01\rangle + |11\rangle - |10\rangle) = |--\rangle$$
$$C_{not}|-+\rangle = C_{not}\tfrac{1}{2}(|00\rangle + |01\rangle - |10\rangle - |11\rangle) = \tfrac{1}{2}(|00\rangle + |01\rangle - |11\rangle - |10\rangle) = |-+\rangle$$
$$C_{not}|--\rangle = C_{not}\tfrac{1}{2}(|00\rangle - |01\rangle - |10\rangle + |11\rangle) = \tfrac{1}{2}(|00\rangle - |01\rangle - |11\rangle + |10\rangle) = |+-\rangle$$

If we then apply the Hadamard transform to each bit the resulting truth table becomes

Initial			Final	
0	0	$\longrightarrow$	0	0

Mermin's interpretation

David Mermin proposed a simpler interpretation for how quantum algorithms and the solution to the Bernstein-Vazirani problem, in particular



Consider a C_{not} acting on the Hadamard basis

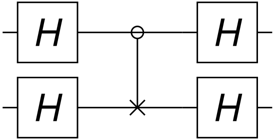
$$C_{not}|++\rangle = C_{not}\tfrac{1}{2}(|00\rangle + |01\rangle + |10\rangle + |11\rangle) = \tfrac{1}{2}(|00\rangle + |01\rangle + |11\rangle + |10\rangle) = |++\rangle$$
$$C_{not}|+-\rangle = C_{not}\tfrac{1}{2}(|00\rangle - |01\rangle + |10\rangle - |11\rangle) = \tfrac{1}{2}(|00\rangle - |01\rangle + |11\rangle - |10\rangle) = |--\rangle$$
$$C_{not}|-+\rangle = C_{not}\tfrac{1}{2}(|00\rangle + |01\rangle - |10\rangle - |11\rangle) = \tfrac{1}{2}(|00\rangle + |01\rangle - |11\rangle - |10\rangle) = |-+\rangle$$
$$C_{not}|--\rangle = C_{not}\tfrac{1}{2}(|00\rangle - |01\rangle - |10\rangle + |11\rangle) = \tfrac{1}{2}(|00\rangle - |01\rangle - |11\rangle + |10\rangle) = |+-\rangle$$

If we then apply the Hadamard transform to each bit the resulting truth table becomes

Initial			Final	
0	0	$\longrightarrow$	0	0
0	1	$\longrightarrow$	1	1

Mermin's interpretation

David Mermin proposed a simpler interpretation for how quantum algorithms and the solution to the Bernstein-Vazirani problem, in particular



Consider a C_{not} acting on the Hadamard basis

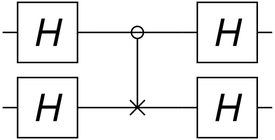
$$C_{not}|++\rangle = C_{not}\tfrac{1}{2}(|00\rangle + |01\rangle + |10\rangle + |11\rangle) = \tfrac{1}{2}(|00\rangle + |01\rangle + |11\rangle + |10\rangle) = |++\rangle$$
$$C_{not}|+-\rangle = C_{not}\tfrac{1}{2}(|00\rangle - |01\rangle + |10\rangle - |11\rangle) = \tfrac{1}{2}(|00\rangle - |01\rangle + |11\rangle - |10\rangle) = |--\rangle$$
$$C_{not}|-+\rangle = C_{not}\tfrac{1}{2}(|00\rangle + |01\rangle - |10\rangle - |11\rangle) = \tfrac{1}{2}(|00\rangle + |01\rangle - |11\rangle - |10\rangle) = |-+\rangle$$
$$C_{not}|--\rangle = C_{not}\tfrac{1}{2}(|00\rangle - |01\rangle - |10\rangle + |11\rangle) = \tfrac{1}{2}(|00\rangle - |01\rangle - |11\rangle + |10\rangle) = |+-\rangle$$

If we then apply the Hadamard transform to each bit the resulting truth table becomes

Initial			Final	
0	0	$\longrightarrow$	0	0
0	1	$\longrightarrow$	1	1
1	0	$\longrightarrow$	1	0

Mermin's interpretation

David Mermin proposed a simpler interpretation for how quantum algorithms and the solution to the Bernstein-Vazirani problem, in particular



Consider a C_{not} acting on the Hadamard basis

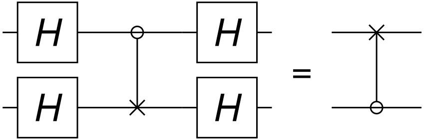
$$C_{not}|++\rangle = C_{not}\tfrac{1}{2}(|00\rangle + |01\rangle + |10\rangle + |11\rangle) = \tfrac{1}{2}(|00\rangle + |01\rangle + |11\rangle + |10\rangle) = |++\rangle$$
$$C_{not}|+-\rangle = C_{not}\tfrac{1}{2}(|00\rangle - |01\rangle + |10\rangle - |11\rangle) = \tfrac{1}{2}(|00\rangle - |01\rangle + |11\rangle - |10\rangle) = |--\rangle$$
$$C_{not}|-+\rangle = C_{not}\tfrac{1}{2}(|00\rangle + |01\rangle - |10\rangle - |11\rangle) = \tfrac{1}{2}(|00\rangle + |01\rangle - |11\rangle - |10\rangle) = |-+\rangle$$
$$C_{not}|--\rangle = C_{not}\tfrac{1}{2}(|00\rangle - |01\rangle - |10\rangle + |11\rangle) = \tfrac{1}{2}(|00\rangle - |01\rangle - |11\rangle + |10\rangle) = |+-\rangle$$

If we then apply the Hadamard transform to each bit the resulting truth table becomes

Initial			Final	
0	0	$\longrightarrow$	0	0
0	1	$\longrightarrow$	1	1
1	0	$\longrightarrow$	1	0
1	1	$\longrightarrow$	0	1

Mermin's interpretation

David Mermin proposed a simpler interpretation for how quantum algorithms and the solution to the Bernstein-Vazirani problem, in particular



Consider a C_{not} acting on the Hadamard basis

$$C_{not}|++\rangle = C_{not}\tfrac{1}{2}(|00\rangle + |01\rangle + |10\rangle + |11\rangle) = \tfrac{1}{2}(|00\rangle + |01\rangle + |11\rangle + |10\rangle) = |++\rangle$$
$$C_{not}|+-\rangle = C_{not}\tfrac{1}{2}(|00\rangle - |01\rangle + |10\rangle - |11\rangle) = \tfrac{1}{2}(|00\rangle - |01\rangle + |11\rangle - |10\rangle) = |--\rangle$$
$$C_{not}|-+\rangle = C_{not}\tfrac{1}{2}(|00\rangle + |01\rangle - |10\rangle - |11\rangle) = \tfrac{1}{2}(|00\rangle + |01\rangle - |11\rangle - |10\rangle) = |-+\rangle$$
$$C_{not}|--\rangle = C_{not}\tfrac{1}{2}(|00\rangle - |01\rangle - |10\rangle + |11\rangle) = \tfrac{1}{2}(|00\rangle - |01\rangle - |11\rangle + |10\rangle) = |+-\rangle$$

If we then apply the Hadamard transform to each bit the resulting truth table becomes

Initial			Final	
0	0	$\longrightarrow$	0	0
0	1	$\longrightarrow$	1	1
1	0	$\longrightarrow$	1	0
1	1	$\longrightarrow$	0	1

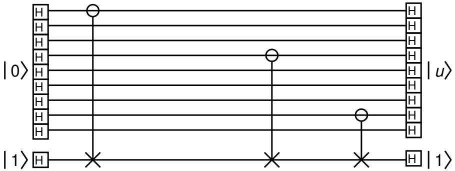
This is simply a C_{not} gate applied to the first qubit controlled by the second

Mermin's interpretation

This insight leads to a simple way to look at the black box for U_{f_u}

Mermin's interpretation

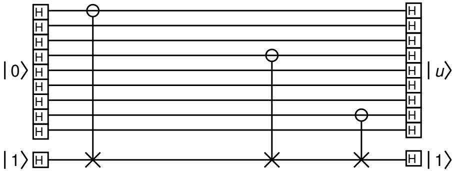
This insight leads to a simple way to look at the black box for U_{f_u}



Mermin's interpretation

This insight leads to a simple way to look at the black box for U_{f_u}

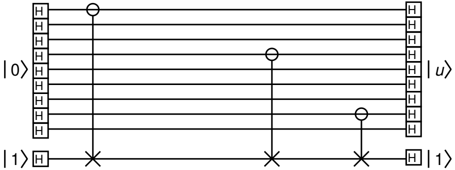
1. Prepare an n-qubit register $|0\rangle_n$



Mermin's interpretation

This insight leads to a simple way to look at the black box for U_{f_u}

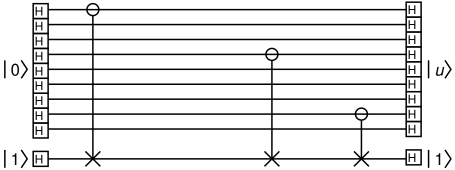
1. Prepare an n-qubit register $|0\rangle_n$
2. Prepare an ancilla qubit $|a\rangle = |1\rangle$



Mermin's interpretation

This insight leads to a simple way to look at the black box for U_{f_u}

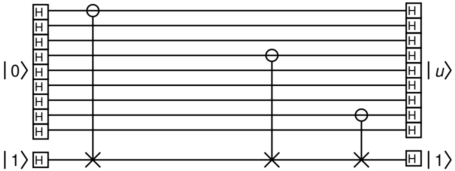
1. Prepare an n-qubit register $|0\rangle_n$
2. Prepare an ancilla qubit $|a\rangle = |1\rangle$
3. Apply the Hadamard gate to all qubits



Mermin's interpretation

This insight leads to a simple way to look at the black box for U_{f_u}

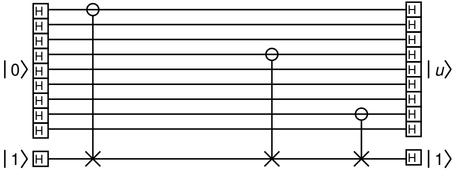
1. Prepare an n-qubit register $|0\rangle_n$
2. Prepare an ancilla qubit $|a\rangle = |1\rangle$
3. Apply the Hadamard gate to all qubits
4. Place a $C_{not}|u_i\rangle|a\rangle$ for each $u_i = 1$



Mermin's interpretation

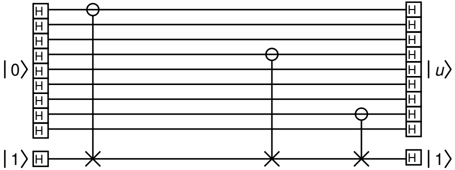
This insight leads to a simple way to look at the black box for U_{f_u}

1. Prepare an n-qubit register $|0\rangle_n$
2. Prepare an ancilla qubit $|a\rangle = |1\rangle$
3. Apply the Hadamard gate to all qubits
4. Place a $C_{not}|u_i\rangle|a\rangle$ for each $u_i = 1$
5. Apply the Hadamard gate to all qubits



Mermin's interpretation

This insight leads to a simple way to look at the black box for U_{f_u}



1. Prepare an n-qubit register $|0\rangle_n$
2. Prepare an ancilla qubit $|a\rangle = |1\rangle$
3. Apply the Hadamard gate to all qubits
4. Place a $C_{not}|u_i\rangle|a\rangle$ for each $u_i = 1$
5. Apply the Hadamard gate to all qubits

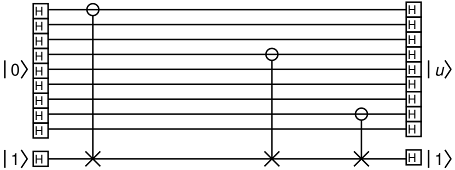
The net effect is to have the ancilla bit "turn on" each qubit in the unknown, $C_{not}|a\rangle|u_i\rangle$ where $u_i = 1$

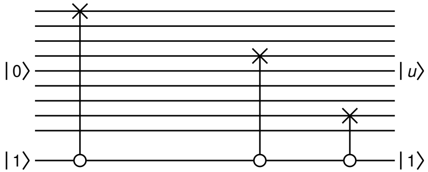
Mermin's interpretation

This insight leads to a simple way to look at the black box for U_{f_u}

1. Prepare an n-qubit register $|0\rangle_n$
2. Prepare an ancilla qubit $|a\rangle = |1\rangle$
3. Apply the Hadamard gate to all qubits
4. Place a $C_{not}|u_i\rangle|a\rangle$ for each $u_i = 1$
5. Apply the Hadamard gate to all qubits

The net effect is to have the ancilla bit "turn on" each qubit in the unknown, $C_{not}|a\rangle|u_i\rangle$ where $u_i = 1$

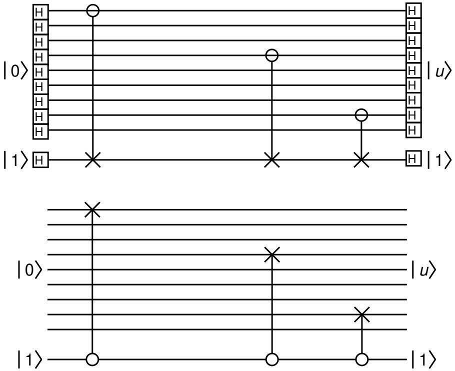




Mermin's interpretation

This insight leads to a simple way to look at the black box for U_{f_u}

1. Prepare an n-qubit register $|0\rangle_n$
2. Prepare an ancilla qubit $|a\rangle = |1\rangle$
3. Apply the Hadamard gate to all qubits
4. Place a $C_{not}|u_i\rangle|a\rangle$ for each $u_i = 1$
5. Apply the Hadamard gate to all qubits



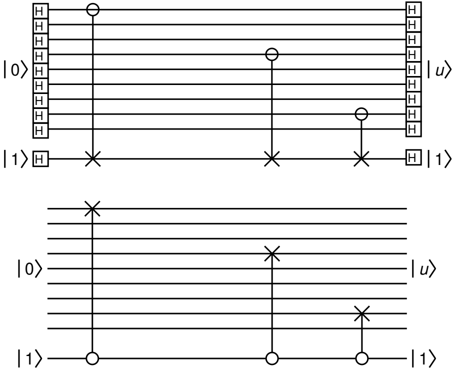
The net effect is to have the ancilla bit "turn on" each qubit in the unknown, $C_{not}|a\rangle|u_i\rangle$ where $u_i = 1$

From this perspective there is no quantum parallelism but simply a discrete circuit which produces the desired outcome

Mermin's interpretation

This insight leads to a simple way to look at the black box for U_{f_u}

1. Prepare an n-qubit register $|0\rangle_n$
2. Prepare an ancilla qubit $|a\rangle = |1\rangle$
3. Apply the Hadamard gate to all qubits
4. Place a $C_{not}|u_i\rangle|a\rangle$ for each $u_i = 1$
5. Apply the Hadamard gate to all qubits



The net effect is to have the ancilla bit "turn on" each qubit in the unknown, $C_{not}|a\rangle|u_i\rangle$ where $u_i = 1$

From this perspective there is no quantum parallelism but simply a discrete circuit which produces the desired outcome

Of course, this presupposes that one knows what $|u\rangle$ is so we are peering into the black box

Simon's problem – description

Suppose we have a 2-to-1 function $f(x)$ such that $f(x) = f(x \oplus a)$ where a is secret and both x and a are n bit strings

Simon's problem – description

Suppose we have a 2-to-1 function $f(x)$ such that $f(x) = f(x \oplus a)$ where a is secret and both x and a are n bit strings

For example, when $n = 3$ we might have the table

Simon's problem – description

Suppose we have a 2-to-1 function $f(x)$ such that $f(x) = f(x \oplus a)$ where a is secret and both x and a are n bit strings

For example, when $n = 3$ we might have the table

There are 4 values for $f(x)$, each appearing twice, once in the top half of the table and once in the bottom

Simon's problem – description

Suppose we have a 2-to-1 function $f(x)$ such that $f(x) = f(x \oplus a)$ where a is secret and both x and a are n bit strings

For example, when $n = 3$ we might have the table

There are 4 values for $f(x)$, each appearing twice, once in the top half of the table and once in the bottom

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

Simon's problem – description

Suppose we have a 2-to-1 function $f(x)$ such that $f(x) = f(x \oplus a)$ where a is secret and both x and a are n bit strings

For example, when $n = 3$ we might have the table

There are 4 values for $f(x)$, each appearing twice, once in the top half of the table and once in the bottom

The goal of the algorithm is to find the the secret string a

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

Simon's problem – description

Suppose we have a 2-to-1 function $f(x)$ such that $f(x) = f(x \oplus a)$ where a is secret and both x and a are n bit strings

For example, when $n = 3$ we might have the table

There are 4 values for $f(x)$, each appearing twice, once in the top half of the table and once in the bottom

The goal of the algorithm is to find the the secret string a

Classically, this can be done by querying the function until we obtain two identical values for $f(x)$ and then calculate $a = x_0 \oplus x_1$

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

Simon's problem – description

Suppose we have a 2-to-1 function $f(x)$ such that $f(x) = f(x \oplus a)$ where a is secret and both x and a are n bit strings

For example, when $n = 3$ we might have the table

There are 4 values for $f(x)$, each appearing twice, once in the top half of the table and once in the bottom

The goal of the algorithm is to find the the secret string a

Classically, this can be done by querying the function until we obtain two identical values for $f(x)$ and then calculate $a = x_0 \oplus x_1$

This can take up to $2^{n-1} + 1$ queries so the computation is $O(2^n)$

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

Simon's problem – description

Suppose we have a 2-to-1 function $f(x)$ such that $f(x) = f(x \oplus a)$ where a is secret and both x and a are n bit strings

For example, when $n = 3$ we might have the table

There are 4 values for $f(x)$, each appearing twice, once in the top half of the table and once in the bottom

The goal of the algorithm is to find the the secret string a

Classically, this can be done by querying the function until we obtain two identical values for $f(x)$ and then calculate $a = x_0 \oplus x_1$

This can take up to $2^{n-1} + 1$ queries so the computation is $O(2^n)$

In contrast, Simon's quantum algorithm is a calculation which is $O(n)$

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

Simon's problem – description

Suppose we have a 2-to-1 function $f(x)$ such that $f(x) = f(x \oplus a)$ where a is secret and both x and a are n bit strings

For example, when $n = 3$ we might have the table

There are 4 values for $f(x)$, each appearing twice, once in the top half of the table and once in the bottom

The goal of the algorithm is to find the the secret string a

Classically, this can be done by querying the function until we obtain two identical values for $f(x)$ and then calculate $a = x_0 \oplus x_1$

This can take up to $2^{n-1} + 1$ queries so the computation is $O(2^n)$

In contrast, Simon's quantum algorithm is a calculation which is $O(n)$

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

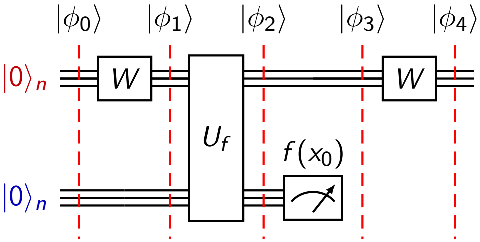
In this case, we can see that $a = 010 \oplus 111 = 101$ and this holds for all matched pairs in the table

Simon's algorithm – quantum circuit

The problem requires two registers of n bits each which we designate with $|0\rangle_n$ and $|0\rangle_n$ as input and output registers, respectively

Simon's algorithm – quantum circuit

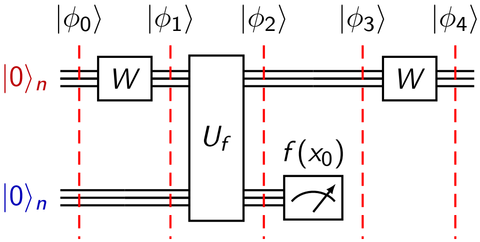
The problem requires two registers of n bits each which we designate with $|0\rangle_n$ and $|0\rangle_n$ as input and output registers, respectively



Simon's algorithm – quantum circuit

The problem requires two registers of n bits each which we designate with $|0\rangle_n$ and $|0\rangle_n$ as input and output registers, respectively

$$|\phi_0\rangle = |0\rangle_n |0\rangle_n$$

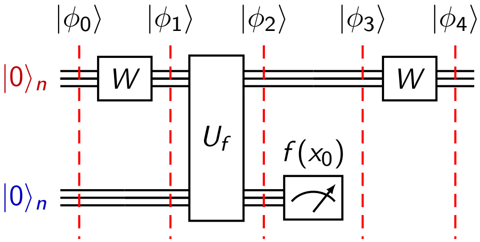


Simon's algorithm – quantum circuit

The problem requires two registers of n bits each which we designate with $|0\rangle_n$ and $|0\rangle_n$ as input and output registers, respectively

$$|\phi_0\rangle = |0\rangle_n|0\rangle_n$$

$$|\phi_1\rangle = W \otimes I(|0\rangle_n|0\rangle_n)$$

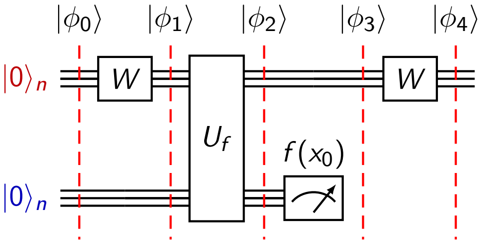


Simon's algorithm – quantum circuit

The problem requires two registers of n bits each which we designate with $|0\rangle_n$ and $|0\rangle_n$ as input and output registers, respectively

$$|\phi_0\rangle = |0\rangle_n|0\rangle_n$$

$$|\phi_1\rangle = W \otimes I(|0\rangle_n|0\rangle_n) = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle|0\rangle_n$$



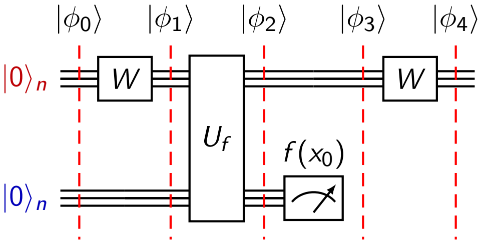
Simon's algorithm – quantum circuit

The problem requires two registers of n bits each which we designate with $|0\rangle_n$ and $|0\rangle_n$ as input and output registers, respectively

$$|\phi_0\rangle = |0\rangle_n |0\rangle_n$$

$$|\phi_1\rangle = W \otimes I(|0\rangle_n |0\rangle_n) = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle |0\rangle_n$$

$$|\phi_2\rangle = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle |f(x)\rangle$$



Simon's algorithm – quantum circuit

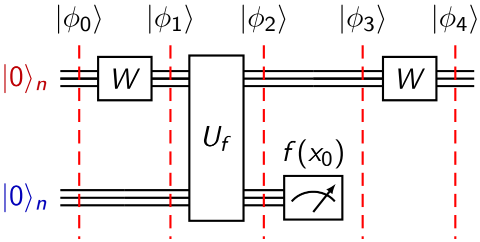
The problem requires two registers of n bits each which we designate with $|0\rangle_n$ and $|0\rangle_n$ as input and output registers, respectively

$$|\phi_0\rangle = |0\rangle_n|0\rangle_n$$

$$|\phi_1\rangle = W \otimes I(|0\rangle_n|0\rangle_n) = \frac{1}{\sqrt{2^n}}\sum_{x=0}^{2^n-1}|x\rangle|0\rangle_n$$

$$|\phi_2\rangle = \frac{1}{\sqrt{2^n}}\sum_{x=0}^{2^n-1}|x\rangle|f(x)\rangle$$

$$|\phi_3\rangle = \frac{1}{\sqrt{2}}\left(|x_0\rangle + |x_0 \oplus a\rangle\right)|f(x_0)\rangle$$



Simon's algorithm – quantum circuit

The problem requires two registers of n bits each which we designate with $|0\rangle_n$ and $|0\rangle_n$ as input and output registers, respectively

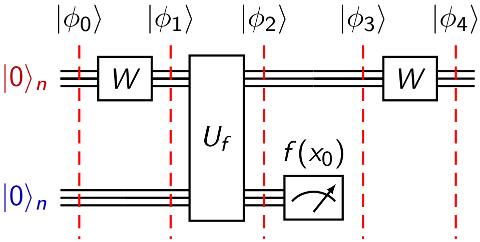
$$|\phi_0\rangle = |0\rangle_n|0\rangle_n$$

$$|\phi_1\rangle = W \otimes I(|0\rangle_n|0\rangle_n) = \frac{1}{\sqrt{2^n}}\sum_{x=0}^{2^n-1}|x\rangle|0\rangle_n$$

$$|\phi_2\rangle = \frac{1}{\sqrt{2^n}}\sum_{x=0}^{2^n-1}|x\rangle|f(x)\rangle$$

$$|\phi_3\rangle = \frac{1}{\sqrt{2}}\left(|x_0\rangle + |x_0 \oplus a\rangle\right)|f(x_0)\rangle$$

$$|\phi_4\rangle = W \otimes I\left[\frac{1}{\sqrt{2}}\left(|x_0\rangle + |x_0 \oplus a\rangle\right)|f(x_0)\rangle\right]$$



Simon's algorithm – quantum circuit

The problem requires two registers of n bits each which we designate with $|0\rangle_n$ and $|0\rangle_n$ as input and output registers, respectively

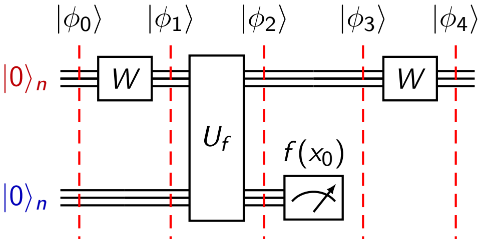
$$|\phi_0\rangle = |0\rangle_n|0\rangle_n$$

$$|\phi_1\rangle = W \otimes I(|0\rangle_n|0\rangle_n) = \frac{1}{\sqrt{2^n}}\sum_{x=0}^{2^n-1}|x\rangle|0\rangle_n$$

$$|\phi_2\rangle = \frac{1}{\sqrt{2^n}}\sum_{x=0}^{2^n-1}|x\rangle|f(x)\rangle$$

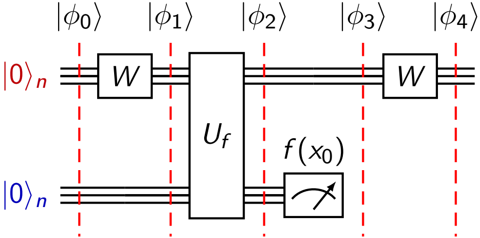
$$|\phi_3\rangle = \frac{1}{\sqrt{2}}\left(|x_0\rangle + |x_0 \oplus a\rangle\right)|f(x_0)\rangle$$

$$|\phi_4\rangle = W \otimes I\left[\frac{1}{\sqrt{2}}\left(|x_0\rangle + |x_0 \oplus a\rangle\right)|f(x_0)\rangle\right] = \frac{1}{\sqrt{2^n}}\frac{1}{\sqrt{2}}\sum_{y=0}^{2^n-1}\left[(-1)^{x_0\cdot y} + (-1)^{(x_0\oplus a)\cdot y}\right]|y\rangle|f(x_0)\rangle$$



Simon's algorithm – quantum circuit

The problem requires two registers of n bits each which we designate with $|0\rangle_n$ and $|0\rangle_n$ as input and output registers, respectively



$$|\phi_0\rangle = |0\rangle_n |0\rangle_n$$

$$|\phi_1\rangle = W \otimes I(|0\rangle_n |0\rangle_n) = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle |0\rangle_n$$

$$|\phi_2\rangle = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle |f(x)\rangle$$

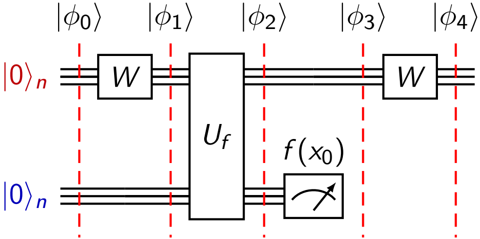
$$|\phi_3\rangle = \frac{1}{\sqrt{2}} \left(|x_0\rangle + |x_0 \oplus a\rangle \right) |f(x_0)\rangle$$

$$|\phi_4\rangle = W \otimes I \left[\frac{1}{\sqrt{2}} \left(|x_0\rangle + |x_0 \oplus a\rangle \right) |f(x_0)\rangle \right] = \frac{1}{\sqrt{2^n}} \frac{1}{\sqrt{2}} \sum_{y=0}^{2^n-1} \left[(-1)^{x_0 \cdot y} + (-1)^{(x_0 \oplus a)\cdot y} \right] |y\rangle |f(x_0)\rangle$$

$$= \frac{1}{\sqrt{2^{n+1}}} \sum_{y=0}^{2^n-1} (-1)^{x_0 \cdot y} \left[1 + (-1)^{a \cdot y} \right] |y\rangle |f(x_0)\rangle$$

Simon's algorithm – quantum circuit

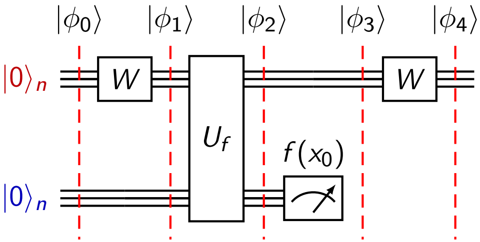
Dropping the $|f(x_0)\rangle$ as it has already been measured, we have



Simon's algorithm – quantum circuit

Dropping the $|f(x_0)\rangle$ as it has already been measured, we have

$$|\phi_4\rangle = \frac{1}{\sqrt{2^{n+1}}} \sum_{y=0}^{2^n-1} (-1)^{x_0 \cdot y} \left[1 + (-1)^{a \cdot y}\right] |y\rangle$$

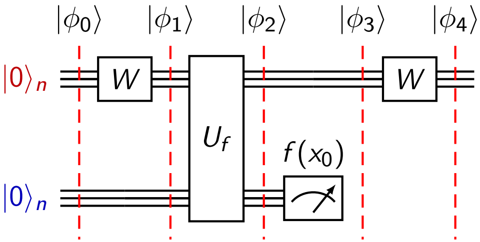


Simon's algorithm – quantum circuit

Dropping the $|f(x_0)\rangle$ as it has already been measured, we have

$$|\phi_4\rangle = \frac{1}{\sqrt{2^{n+1}}} \sum_{y=0}^{2^n-1} (-1)^{x_0 \cdot y} \left[1 + (-1)^{a \cdot y}\right] |y\rangle$$

There are two cases to consider for the modulo 2 scalar product $a \cdot y$



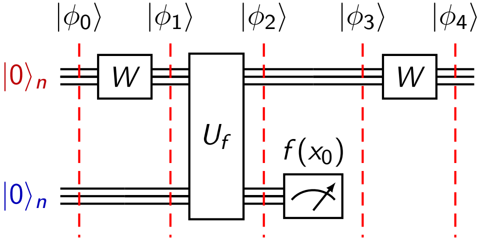
Simon's algorithm – quantum circuit

Dropping the $|f(x_0)\rangle$ as it has already been measured, we have

$$|\phi_4\rangle = \frac{1}{\sqrt{2^{n+1}}} \sum_{y=0}^{2^n-1} (-1)^{x_0 \cdot y} \left[1 + (-1)^{a \cdot y}\right] |y\rangle$$

There are two cases to consider for the modulo 2 scalar product $a \cdot y$

$$y \cdot a \neq 0 \quad \longrightarrow \quad |\phi_4\rangle \equiv 0$$



Simon's algorithm – quantum circuit

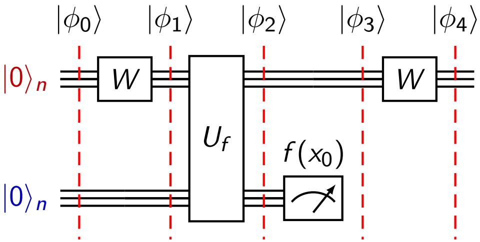
Dropping the $|f(x_0)\rangle$ as it has already been measured, we have

$$|\phi_4\rangle = \frac{1}{\sqrt{2^{n+1}}} \sum_{y=0}^{2^n-1} (-1)^{x_0 \cdot y} \left[1 + (-1)^{a \cdot y}\right] |y\rangle$$

There are two cases to consider for the modulo 2 scalar product $a \cdot y$

$$y \cdot a \neq 0 \quad \longrightarrow \quad |\phi_4\rangle \equiv 0$$

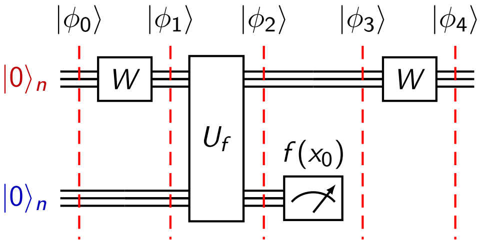
The second case is for $a \cdot y = 0$, in which case



Simon's algorithm – quantum circuit

Dropping the $|f(x_0)\rangle$ as it has already been measured, we have

$$|\phi_4\rangle = \frac{1}{\sqrt{2^{n+1}}} \sum_{y=0}^{2^n-1} (-1)^{x_0 \cdot y} \left[1 + (-1)^{a \cdot y}\right] |y\rangle$$



There are two cases to consider for the modulo 2 scalar product $a \cdot y$

$$y \cdot a \neq 0 \quad \longrightarrow \quad |\phi_4\rangle \equiv 0$$

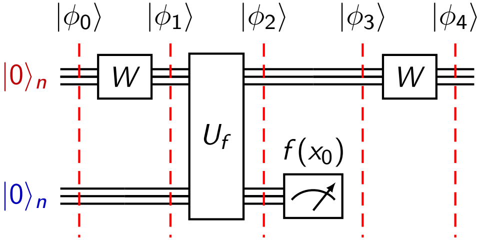
The second case is for $a \cdot y = 0$, in which case

$$|\phi_4\rangle = \frac{1}{\sqrt{2^{n+1}}} \sum_{y=0}^{2^n-1} (-1)^{x_0 \cdot y} \left[1 + 1\right] |y\rangle$$

Simon's algorithm – quantum circuit

Dropping the $|f(x_0)\rangle$ as it has already been measured, we have

$$|\phi_4\rangle = \frac{1}{\sqrt{2^{n+1}}} \sum_{y=0}^{2^n-1} (-1)^{x_0 \cdot y} \left[1 + (-1)^{a \cdot y}\right] |y\rangle$$



There are two cases to consider for the modulo 2 scalar product $a \cdot y$

$$y \cdot a \neq 0 \quad \longrightarrow \quad |\phi_4\rangle \equiv 0$$

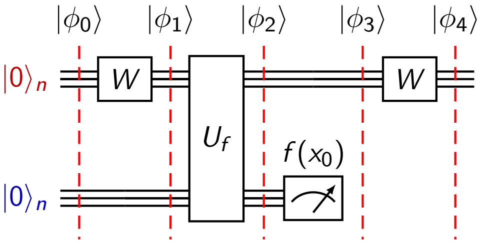
The second case is for $a \cdot y = 0$, in which case

$$|\phi_4\rangle = \frac{1}{\sqrt{2^{n+1}}} \sum_{y=0}^{2^n-1} (-1)^{x_0 \cdot y} \left[1 + 1\right] |y\rangle = \frac{1}{\sqrt{2^{n-1}}} \sum_{y=0}^{2^n-1} (-1)^{x_0 \cdot y} |y\rangle$$

Simon's algorithm – quantum circuit

Dropping the $|f(x_0)\rangle$ as it has already been measured, we have

$$|\phi_4\rangle = \frac{1}{\sqrt{2^{n+1}}} \sum_{y=0}^{2^n-1} (-1)^{x_0 \cdot y} \left[1 + (-1)^{a \cdot y}\right] |y\rangle$$



There are two cases to consider for the modulo 2 scalar product $a \cdot y$

$$y \cdot a \neq 0 \quad \longrightarrow \quad |\phi_4\rangle \equiv 0$$

The second case is for $a \cdot y = 0$, in which case

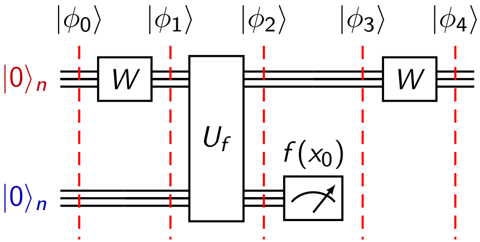
$$|\phi_4\rangle = \frac{1}{\sqrt{2^{n+1}}} \sum_{y=0}^{2^n-1} (-1)^{x_0 \cdot y} [1+1] |y\rangle = \frac{1}{\sqrt{2^{n-1}}} \sum_{y=0}^{2^n-1} (-1)^{x_0 \cdot y} |y\rangle$$

This is a superposition of 2^n possible states, one of which will be observed when $|\phi_4\rangle$ is measured

Simon's algorithm – quantum circuit

Dropping the $|f(x_0)\rangle$ as it has already been measured, we have

$$|\phi_4\rangle = \frac{1}{\sqrt{2^{n+1}}}\sum_{y=0}^{2^n-1}(-1)^{x_0\cdot y}\left[1+(-1)^{a\cdot y}\right]|y\rangle$$



There are two cases to consider for the modulo 2 scalar product $a \cdot y$

$$y \cdot a \neq 0 \quad \longrightarrow \quad |\phi_4\rangle \equiv 0$$

The second case is for $a \cdot y = 0$, in which case

$$|\phi_4\rangle = \frac{1}{\sqrt{2^{n+1}}}\sum_{y=0}^{2^n-1}(-1)^{x_0\cdot y}\left[1+1\right]|y\rangle = \frac{1}{\sqrt{2^{n-1}}}\sum_{y=0}^{2^n-1}(-1)^{x_0\cdot y}|y\rangle$$

This is a superposition of 2^n possible states, one of which will be observed when $|\phi_4\rangle$ is measured

If $n-1$ linearly independent $|y\rangle$ are measured, it is possible to solve $y \cdot a = 0$

Simon's algorithm – example

Suppose a system with $n = 4$ and $a = 1001$, $f(x)$ has the truth table

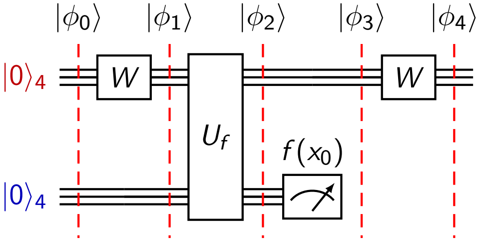
Simon's algorithm – example

Suppose a system with $n = 4$ and $a = 1001$, $f(x)$ has the truth table

x	$f(x)$
0000	1111
0001	0001
0010	1110
0011	1101
0100	0000
0101	0101
0110	1010
0111	1001
1000	0001
1001	1111
1010	1101
1011	1110
1100	0101
1101	0000
1110	1001
1111	1010

Simon's algorithm – example

Suppose a system with $n = 4$ and $a = 1001$, $f(x)$ has the truth table

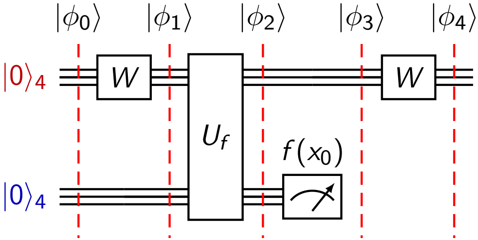


x	$f(x)$
0000	1111
0001	0001
0010	1110
0011	1101
0100	0000
0101	0101
0110	1010
0111	1001
1000	0001
1001	1111
1010	1101
1011	1110
1100	0101
1101	0000
1110	1001
1111	1010

Simon's algorithm – example

Suppose a system with $n = 4$ and $a = 1001$, $f(x)$ has the truth table

$|\phi_0\rangle = |0\rangle|0\rangle = |0000\rangle|0000\rangle$



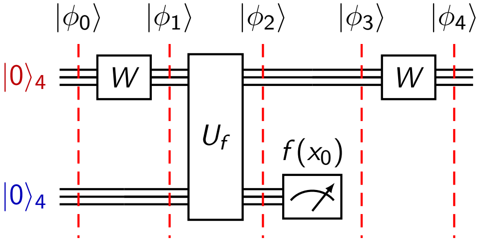
x	$f(x)$
0000	1111
0001	0001
0010	1110
0011	1101
0100	0000
0101	0101
0110	1010
0111	1001
1000	0001
1001	1111
1010	1101
1011	1110
1100	0101
1101	0000
1110	1001
1111	1010

Simon's algorithm – example

Suppose a system with $n = 4$ and $a = 1001$, $f(x)$ has the truth table

$$|\phi_0\rangle = |0\rangle|0\rangle = |0000\rangle|0000\rangle$$

$$|\phi_1\rangle = \frac{1}{4}\sum_{x=0}^{15}|x\rangle|0000\rangle$$



x	$f(x)$
0000	1111
0001	0001
0010	1110
0011	1101
0100	0000
0101	0101
0110	1010
0111	1001
1000	0001
1001	1111
1010	1101
1011	1110
1100	0101
1101	0000
1110	1001
1111	1010

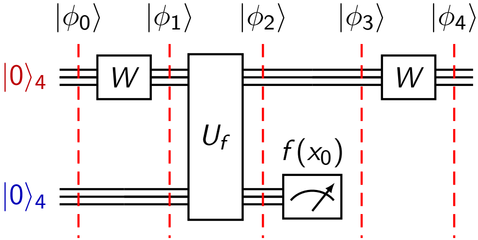
Simon's algorithm – example

Suppose a system with $n = 4$ and $a = 1001$, $f(x)$ has the truth table

$$|\phi_0\rangle = |0\rangle|0\rangle = |0000\rangle|0000\rangle$$

$$|\phi_1\rangle = \frac{1}{4}\sum_{x=0}^{15}|x\rangle|0000\rangle$$

$$|\phi_2\rangle = \frac{1}{4}\sum_{x=0}^{15}|x\rangle|f(x)\rangle$$



x	$f(x)$
0000	1111
0001	0001
0010	1110
0011	1101
0100	0000
0101	0101
0110	1010
0111	1001
1000	0001
1001	1111
1010	1101
1011	1110
1100	0101
1101	0000
1110	1001
1111	1010

Simon's algorithm – example

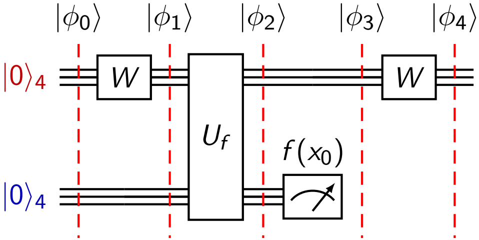
Suppose a system with $n = 4$ and $a = 1001$, $f(x)$ has the truth table

$$|\phi_0\rangle = |0\rangle|0\rangle = |0000\rangle|0000\rangle$$

$$|\phi_1\rangle = \frac{1}{4}\sum_{x=0}^{15} |x\rangle|0000\rangle$$

$$|\phi_2\rangle = \frac{1}{4}\sum_{x=0}^{15} |x\rangle|f(x)\rangle$$

$$|\phi_3\rangle = \frac{1}{\sqrt{2}}\left[|x_0\rangle + |x_0 \oplus a\rangle\right]|f(x_0)\rangle$$



x	$f(x)$
0000	1111
0001	0001
0010	1110
0011	1101
0100	0000
0101	0101
0110	1010
0111	1001
1000	0001
1001	1111
1010	1101
1011	1110
1100	0101
1101	0000
1110	1001
1111	1010

Simon's algorithm – example

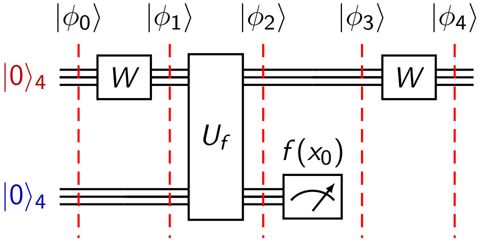
Suppose a system with $n = 4$ and $a = 1001$, $f(x)$ has the truth table

$$|\phi_0\rangle = |0\rangle|0\rangle = |0000\rangle|0000\rangle$$

$$|\phi_1\rangle = \frac{1}{4}\sum_{x=0}^{15}|x\rangle|0000\rangle$$

$$|\phi_2\rangle = \frac{1}{4}\sum_{x=0}^{15}|x\rangle|f(x)\rangle$$

$$|\phi_3\rangle = \frac{1}{\sqrt{2}}\left[|x_0\rangle + |x_0 \oplus a\rangle\right]|f(x_0)\rangle$$



For example, suppose $f(x_0) = 1010$

x	$f(x)$
0000	1111
0001	0001
0010	1110
0011	1101
0100	0000
0101	0101
0110	1010
0111	1001
1000	0001
1001	1111
1010	1101
1011	1110
1100	0101
1101	0000
1110	1001
1111	1010

Simon's algorithm – example

Suppose a system with $n = 4$ and $a = 1001$, $f(x)$ has the truth table

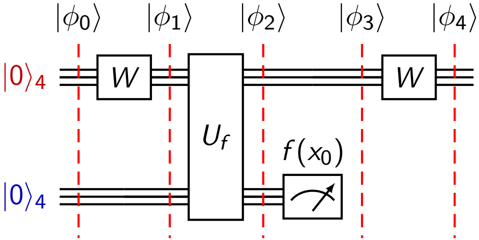
$$|\phi_0\rangle = |0\rangle|0\rangle = |0000\rangle|0000\rangle$$

$$|\phi_1\rangle = \frac{1}{4}\sum_{x=0}^{15}|x\rangle|0000\rangle$$

$$|\phi_2\rangle = \frac{1}{4}\sum_{x=0}^{15}|x\rangle|f(x)\rangle$$

$$|\phi_3\rangle = \frac{1}{\sqrt{2}}\left[|x_0\rangle + |x_0 \oplus a\rangle\right]|f(x_0)\rangle$$

$$|\phi_3\rangle = \frac{\left[|0110\rangle + |1111\rangle\right]}{\sqrt{2}}|f(x_0)\rangle$$



For example, suppose $f(x_0) = 1010$

x	$f(x)$
0000	1111
0001	0001
0010	1110
0011	1101
0100	0000
0101	0101
0110	1010
0111	1001
1000	0001
1001	1111
1010	1101
1011	1110
1100	0101
1101	0000
1110	1001
1111	1010

Simon's algorithm – example

Suppose a system with $n = 4$ and $a = 1001$, $f(x)$ has the truth table

x	$f(x)$
0000	1111
0001	0001
0010	1110
0011	1101
0100	0000
0101	0101
0110	1010
0111	1001
1000	0001
1001	1111
1010	1101
1011	1110
1100	0101
1101	0000
1110	1001
1111	1010

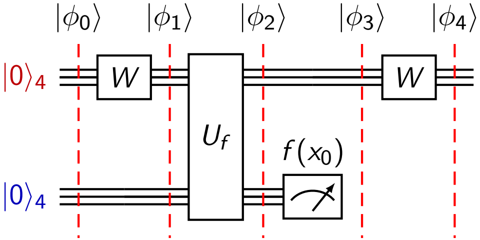
$$|\phi_0\rangle = |0\rangle|0\rangle = |0000\rangle|0000\rangle$$

$$|\phi_1\rangle = \frac{1}{4}\sum_{x=0}^{15}|x\rangle|0000\rangle$$

$$|\phi_2\rangle = \frac{1}{4}\sum_{x=0}^{15}|x\rangle|f(x)\rangle$$

$$|\phi_3\rangle = \frac{1}{\sqrt{2}}\left[|x_0\rangle + |x_0 \oplus a\rangle\right]|f(x_0)\rangle$$

$$|\phi_3\rangle = \frac{[|0110\rangle + |1111\rangle]}{\sqrt{2}} \cancel{|f(x_0)\rangle}$$



For example, suppose $f(x_0) = 1010$

Simon's algorithm – example

Suppose a system with $n = 4$ and $a = 1001$, $f(x)$ has the truth table

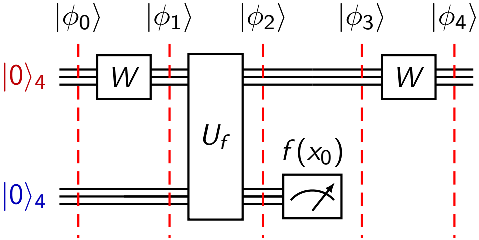
x	$f(x)$
0000	1111
0001	0001
0010	1110
0011	1101
0100	0000
0101	0101
0110	1010
0111	1001
1000	0001
1001	1111
1010	1101
1011	1110
1100	0101
1101	0000
1110	1001
1111	1010

$$|\phi_0\rangle = |0\rangle|0\rangle = |0000\rangle|0000\rangle$$

$$|\phi_1\rangle = \frac{1}{4}\sum_{x=0}^{15}|x\rangle|0000\rangle$$

$$|\phi_2\rangle = \frac{1}{4}\sum_{x=0}^{15}|x\rangle|f(x)\rangle$$

$$|\phi_3\rangle = \frac{1}{\sqrt{2}}\left[|x_0\rangle + |x_0 \oplus a\rangle\right]|f(x_0)\rangle$$



For example, suppose $f(x_0) = 1010$

$$|\phi_3\rangle = \frac{[|0110\rangle + |1111\rangle]}{\sqrt{2}}\cancel{|f(x_0)\rangle}$$

now apply the Walsh transformation

Simon's algorithm – example

Suppose a system with $n = 4$ and $a = 1001$, $f(x)$ has the truth table

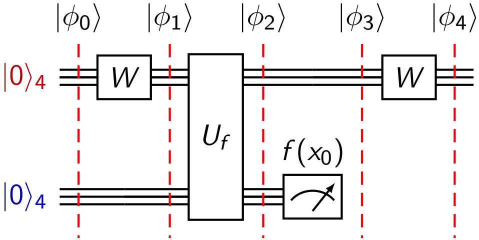
x	$f(x)$
0000	1111
0001	0001
0010	1110
0011	1101
0100	0000
0101	0101
0110	1010
0111	1001
1000	0001
1001	1111
1010	1101
1011	1110
1100	0101
1101	0000
1110	1001
1111	1010

$$|\phi_0\rangle = |0\rangle|0\rangle = |0000\rangle|0000\rangle$$

$$|\phi_1\rangle = \frac{1}{4}\sum_{x=0}^{15}|x\rangle|0000\rangle$$

$$|\phi_2\rangle = \frac{1}{4}\sum_{x=0}^{15}|x\rangle|f(x)\rangle$$

$$|\phi_3\rangle = \frac{1}{\sqrt{2}}\left[|x_0\rangle + |x_0 \oplus a\rangle\right]|f(x_0)\rangle$$



For example, suppose $f(x_0) = 1010$

$$|\phi_3\rangle = \frac{[|0110\rangle + |1111\rangle]}{\sqrt{2}}\cancel{|f(x_0)\rangle}$$ now apply the Walsh transformation

$$|\phi_4\rangle = \frac{[|0000\rangle - |0010\rangle - |0100\rangle + |0110\rangle + |1001\rangle - |1011\rangle - |1101\rangle + |1111\rangle]}{\sqrt{8}}$$

Simon's algorithm – example

Suppose a system with $n=4$ and $a=1001$, $f(x)$ has the truth table

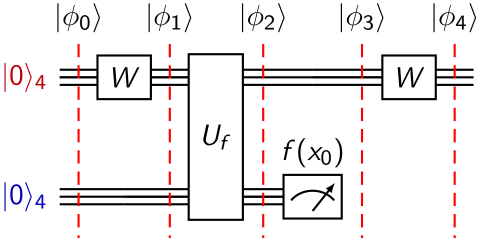
x	$f(x)$
0000	1111
0001	0001
0010	1110
0011	1101
0100	0000
0101	0101
0110	1010
0111	1001
1000	0001
1001	1111
1010	1101
1011	1110
1100	0101
1101	0000
1110	1001
1111	1010

$$|\phi_0\rangle = |0\rangle|0\rangle = |0000\rangle|0000\rangle$$

$$|\phi_1\rangle = \frac{1}{4}\sum_{x=0}^{15}|x\rangle|0000\rangle$$

$$|\phi_2\rangle = \frac{1}{4}\sum_{x=0}^{15}|x\rangle|f(x)\rangle$$

$$|\phi_3\rangle = \frac{1}{\sqrt{2}}\left[|x_0\rangle + |x_0 \oplus a\rangle\right]|f(x_0)\rangle$$



For example, suppose $f(x_0) = 1010$

$$|\phi_3\rangle = \frac{[|0110\rangle + |1111\rangle]}{\sqrt{2}}\cancel{|f(x_0)\rangle}$$ now apply the Walsh transformation

$$|\phi_4\rangle = \frac{[|0000\rangle - |0010\rangle - |0100\rangle + |0110\rangle + |1001\rangle - |1011\rangle - |1101\rangle + |1111\rangle]}{\sqrt{8}}$$

Note that any value of $|f(x_0)\rangle$ measured will result in these 8 $|x_0\rangle$

Simon's algorithm – example

$$|\phi_4\rangle = \frac{1}{\sqrt{8}}\left[|0000\rangle - |0010\rangle - |0100\rangle + |0110\rangle + |1001\rangle - |1011\rangle - |1101\rangle + |1111\rangle\right]$$

Simon's algorithm – example

$$|\phi_4\rangle = \frac{1}{\sqrt{8}}\left[|0000\rangle - |0010\rangle - |0100\rangle + |0110\rangle + |1001\rangle - |1011\rangle - |1101\rangle + |1111\rangle\right]$$

The result of the final measurement, $|y\rangle$ will be one of these eight values and each of them should satisfy the linear equation $a \cdot y = 0$

Simon's algorithm – example

$$|\phi_4\rangle = \frac{1}{\sqrt{8}}\left[|0000\rangle - |0010\rangle - |0100\rangle + |0110\rangle + |1001\rangle - |1011\rangle - |1101\rangle + |1111\rangle\right]$$

The result of the final measurement, $|y\rangle$ will be one of these eight values and each of them should satisfy the linear equation $a \cdot y = 0$

Since we know that $a = |1001\rangle$ for this example, we can check this identity

Simon's algorithm – example

$$|\phi_4\rangle = \frac{1}{\sqrt{8}}\left[|0000\rangle - |0010\rangle - |0100\rangle + |0110\rangle + |1001\rangle - |1011\rangle - |1101\rangle + |1111\rangle\right]$$

The result of the final measurement, $|y\rangle$ will be one of these eight values and each of them should satisfy the linear equation $a \cdot y = 0$

Since we know that $a = |1001\rangle$ for this example, we can check this identity

$$|1001\rangle \cdot |0000\rangle = 1{\cdot}0 + 0{\cdot}0 + 0{\cdot}0 + 1{\cdot}0 = 0$$

Simon's algorithm – example

$$|\phi_4\rangle = \frac{1}{\sqrt{8}}\left[|0000\rangle - |0010\rangle - |0100\rangle + |0110\rangle + |1001\rangle - |1011\rangle - |1101\rangle + |1111\rangle\right]$$

The result of the final measurement, $|y\rangle$ will be one of these eight values and each of them should satisfy the linear equation $a \cdot y = 0$

Since we know that $a = |1001\rangle$ for this example, we can check this identity

$$|1001\rangle \cdot |0000\rangle = 1{\cdot}0 + 0{\cdot}0 + 0{\cdot}0 + 1{\cdot}0 = 0$$

$$|1001\rangle \cdot |1001\rangle = 1{\cdot}1 + 0{\cdot}0 + 0{\cdot}0 + 1{\cdot}1 = 2$$

Simon's algorithm – example

$$|\phi_4\rangle = \frac{1}{\sqrt{8}}\left[|0000\rangle - |0010\rangle - |0100\rangle + |0110\rangle + |1001\rangle - |1011\rangle - |1101\rangle + |1111\rangle\right]$$

The result of the final measurement, $|y\rangle$ will be one of these eight values and each of them should satisfy the linear equation $a \cdot y = 0$

Since we know that $a = |1001\rangle$ for this example, we can check this identity

$$|1001\rangle \cdot |0000\rangle = 1 \cdot 0 + 0 \cdot 0 + 0 \cdot 0 + 1 \cdot 0 = 0$$

$$|1001\rangle \cdot |1001\rangle = 1 \cdot 1 + 0 \cdot 0 + 0 \cdot 0 + 1 \cdot 1 = 2 = 0$$

Simon's algorithm – example

$$|\phi_4\rangle = \frac{1}{\sqrt{8}}\left[|0000\rangle - |0010\rangle - |0100\rangle + |0110\rangle + |1001\rangle - |1011\rangle - |1101\rangle + |1111\rangle\right]$$

The result of the final measurement, $|y\rangle$ will be one of these eight values and each of them should satisfy the linear equation $a \cdot y = 0$

Since we know that $a = |1001\rangle$ for this example, we can check this identity

$$|1001\rangle \cdot |0000\rangle = 1\cdot 0 + 0\cdot 0 + 0\cdot 0 + 1\cdot 0 = 0$$

$$|1001\rangle \cdot |1001\rangle = 1\cdot 1 + 0\cdot 0 + 0\cdot 0 + 1\cdot 1 = 2 = 0$$

and the other 6 have the same properties

It is now necessary to collect $n - 1 = 3$ independent values of $|y\rangle$ to solve for a

Simon's algorithm – example

$$|\phi_4\rangle = \frac{1}{\sqrt{8}}\left[|0000\rangle - |0010\rangle - |0100\rangle + |0110\rangle + |1001\rangle - |1011\rangle - |1101\rangle + |1111\rangle\right]$$

The result of the final measurement, $|y\rangle$ will be one of these eight values and each of them should satisfy the linear equation $a \cdot y = 0$

Since we know that $a = |1001\rangle$ for this example, we can check this identity

$$|1001\rangle \cdot |0000\rangle = 1\cdot 0 + 0\cdot 0 + 0\cdot 0 + 1\cdot 0 = 0$$

$$|1001\rangle \cdot |1001\rangle = 1\cdot 1 + 0\cdot 0 + 0\cdot 0 + 1\cdot 1 = 2 = 0$$

and the other 6 have the same properties

It is now necessary to collect $n - 1 = 3$ independent values of $|y\rangle$ to solve for a

Trial	$\lvert y\rangle$	Indep.?

Simon's algorithm – example

$$|\phi_4\rangle = \frac{1}{\sqrt{8}}\left[|0000\rangle - |0010\rangle - |0100\rangle + |0110\rangle + |1001\rangle - |1011\rangle - |1101\rangle + |1111\rangle\right]$$

The result of the final measurement, $|y\rangle$ will be one of these eight values and each of them should satisfy the linear equation $a \cdot y = 0$

Since we know that $a = |1001\rangle$ for this example, we can check this identity

$$|1001\rangle \cdot |0000\rangle = 1\cdot 0 + 0\cdot 0 + 0\cdot 0 + 1\cdot 0 = 0$$

$$|1001\rangle \cdot |1001\rangle = 1\cdot 1 + 0\cdot 0 + 0\cdot 0 + 1\cdot 1 = 2 = 0$$

and the other 6 have the same properties

It is now necessary to collect $n - 1 = 3$ independent values of $|y\rangle$ to solve for a

Trial	$\vert y\rangle$	Indep.?
1	$\vert 0000\rangle$	No

Simon's algorithm – example

$$|\phi_4\rangle = \frac{1}{\sqrt{8}}\left[|0000\rangle - |0010\rangle - |0100\rangle + |0110\rangle + |1001\rangle - |1011\rangle - |1101\rangle + |1111\rangle\right]$$

The result of the final measurement, $|y\rangle$ will be one of these eight values and each of them should satisfy the linear equation $a \cdot y = 0$

Since we know that $a = |1001\rangle$ for this example, we can check this identity

$$|1001\rangle \cdot |0000\rangle = 1\cdot 0 + 0\cdot 0 + 0\cdot 0 + 1\cdot 0 = 0$$

$$|1001\rangle \cdot |1001\rangle = 1\cdot 1 + 0\cdot 0 + 0\cdot 0 + 1\cdot 1 = 2 = 0$$

and the other 6 have the same properties

It is now necessary to collect $n - 1 = 3$ independent values of $|y\rangle$ to solve for a

Trial	$\|y\rangle$	Indep.?
1	$\|0000\rangle$	No
1	$\|0010\rangle$	Yes

Simon's algorithm – example

$$|\phi_4\rangle = \frac{1}{\sqrt{8}}\left[|0000\rangle - |0010\rangle - |0100\rangle + |0110\rangle + |1001\rangle - |1011\rangle - |1101\rangle + |1111\rangle\right]$$

The result of the final measurement, $|y\rangle$ will be one of these eight values and each of them should satisfy the linear equation $a \cdot y = 0$

Since we know that $a = |1001\rangle$ for this example, we can check this identity

$$|1001\rangle \cdot |0000\rangle = 1\cdot 0 + 0\cdot 0 + 0\cdot 0 + 1\cdot 0 = 0$$

$$|1001\rangle \cdot |1001\rangle = 1\cdot 1 + 0\cdot 0 + 0\cdot 0 + 1\cdot 1 = 2 = 0$$

and the other 6 have the same properties

It is now necessary to collect $n - 1 = 3$ independent values of $|y\rangle$ to solve for a

Trial	$\|y\rangle$	Indep.?
1	$\|0000\rangle$	No
1	$\|0010\rangle$	Yes
1	$\|0100\rangle$	Yes

Simon's algorithm – example

$$|\phi_4\rangle = \frac{1}{\sqrt{8}}\left[|0000\rangle - |0010\rangle - |0100\rangle + |0110\rangle + |1001\rangle - |1011\rangle - |1101\rangle + |1111\rangle\right]$$

The result of the final measurement, $|y\rangle$ will be one of these eight values and each of them should satisfy the linear equation $a \cdot y = 0$

Since we know that $a = |1001\rangle$ for this example, we can check this identity

$$|1001\rangle \cdot |0000\rangle = 1 \cdot 0 + 0 \cdot 0 + 0 \cdot 0 + 1 \cdot 0 = 0$$

$$|1001\rangle \cdot |1001\rangle = 1 \cdot 1 + 0 \cdot 0 + 0 \cdot 0 + 1 \cdot 1 = 2 = 0$$

and the other 6 have the same properties

It is now necessary to collect $n - 1 = 3$ independent values of $|y\rangle$ to solve for a

Trial	$\lvert y\rangle$	Indep.?
1	$\lvert 0000\rangle$	No
1	$\lvert 0010\rangle$	Yes
1	$\lvert 0100\rangle$	Yes
1	$\lvert 0110\rangle$	No

Simon's algorithm – example

$$|\phi_4\rangle = \frac{1}{\sqrt{8}}\left[|0000\rangle - |0010\rangle - |0100\rangle + |0110\rangle + |1001\rangle - |1011\rangle - |1101\rangle + |1111\rangle\right]$$

The result of the final measurement, $|y\rangle$ will be one of these eight values and each of them should satisfy the linear equation $a \cdot y = 0$

Since we know that $a = |1001\rangle$ for this example, we can check this identity

$$|1001\rangle \cdot |0000\rangle = 1\cdot 0 + 0\cdot 0 + 0\cdot 0 + 1\cdot 0 = 0$$

$$|1001\rangle \cdot |1001\rangle = 1\cdot 1 + 0\cdot 0 + 0\cdot 0 + 1\cdot 1 = 2 = 0$$

and the other 6 have the same properties

It is now necessary to collect $n - 1 = 3$ independent values of $|y\rangle$ to solve for a

Trial	$\|y\rangle$	Indep.?
1	$\|0000\rangle$	No
1	$\|0010\rangle$	Yes
1	$\|0100\rangle$	Yes
1	$\|0110\rangle$	No
1	$\|1001\rangle$	Yes

Simon's algorithm – example

$$|\phi_4\rangle = \frac{1}{\sqrt{8}}\left[|0000\rangle - |0010\rangle - |0100\rangle + |0110\rangle + |1001\rangle - |1011\rangle - |1101\rangle + |1111\rangle\right]$$

The result of the final measurement, $|y\rangle$ will be one of these eight values and each of them should satisfy the linear equation $a \cdot y = 0$

Since we know that $a = |1001\rangle$ for this example, we can check this identity

$$|1001\rangle \cdot |0000\rangle = 1\cdot 0 + 0\cdot 0 + 0\cdot 0 + 1\cdot 0 = 0$$

$$|1001\rangle \cdot |1001\rangle = 1\cdot 1 + 0\cdot 0 + 0\cdot 0 + 1\cdot 1 = 2 = 0$$

and the other 6 have the same properties

It is now necessary to collect $n - 1 = 3$ independent values of $|y\rangle$ to solve for a

Trial	$\lvert y\rangle$	Indep.?
1	$\lvert 0000\rangle$	No
1	$\lvert 0010\rangle$	Yes
1	$\lvert 0100\rangle$	Yes
1	$\lvert 0110\rangle$	No
1	$\lvert 1001\rangle$	Yes

Create a matrix from the $y \cdot a = 0$ equation and the three independent values obtained

Simon's algorithm – example

$$|\phi_4\rangle = \frac{1}{\sqrt{8}}\left[|0000\rangle - |0010\rangle - |0100\rangle + |0110\rangle + |1001\rangle - |1011\rangle - |1101\rangle + |1111\rangle\right]$$

The result of the final measurement, $|y\rangle$ will be one of these eight values and each of them should satisfy the linear equation $a \cdot y = 0$

Since we know that $a = |1001\rangle$ for this example, we can check this identity

and the other 6 have the same properties

$$|1001\rangle\cdot|0000\rangle = 1\cdot0 + 0\cdot0 + 0\cdot0 + 1\cdot0 = 0$$

$$|1001\rangle\cdot|1001\rangle = 1\cdot1 + 0\cdot0 + 0\cdot0 + 1\cdot1 = 2 = 0$$

It is now necessary to collect $n - 1 = 3$ independent values of $|y\rangle$ to solve for a

Trial	$\|y\rangle$	Indep.?
1	$\|0000\rangle$	No
1	$\|0010\rangle$	Yes
1	$\|0100\rangle$	Yes
1	$\|0110\rangle$	No
1	$\|1001\rangle$	Yes

Create a matrix from the $y \cdot a = 0$ equation and the three independent values obtained

$$\begin{bmatrix} & & & \\ & & & \\ & & & \end{bmatrix}\begin{bmatrix} \\ \\ \\ \end{bmatrix} = \begin{bmatrix} \\ \\ \end{bmatrix}$$

Simon's algorithm – example

$$|\phi_4\rangle = \frac{1}{\sqrt{8}}\left[|0000\rangle - |0010\rangle - |0100\rangle + |0110\rangle + |1001\rangle - |1011\rangle - |1101\rangle + |1111\rangle\right]$$

The result of the final measurement, $|y\rangle$ will be one of these eight values and each of them should satisfy the linear equation $a \cdot y = 0$

Since we know that $a = |1001\rangle$ for this example, we can check this identity

$$|1001\rangle \cdot |0000\rangle = 1{\cdot}0 + 0{\cdot}0 + 0{\cdot}0 + 1{\cdot}0 = 0$$

$$|1001\rangle \cdot |1001\rangle = 1{\cdot}1 + 0{\cdot}0 + 0{\cdot}0 + 1{\cdot}1 = 2 = 0$$

and the other 6 have the same properties

It is now necessary to collect $n - 1 = 3$ independent values of $|y\rangle$ to solve for a

Trial	$\|y\rangle$	Indep.?
1	$\|0000\rangle$	No
1	$\|0010\rangle$	Yes
1	$\|0100\rangle$	Yes
1	$\|0110\rangle$	No
1	$\|1001\rangle$	Yes

Create a matrix from the $y \cdot a = 0$ equation and the three independent values obtained

$$\begin{bmatrix} 0 & 0 & 1 & 0 \\ & & & \\ & & & \end{bmatrix} \begin{bmatrix} a_3 \\ \\ \\ \end{bmatrix} = \begin{bmatrix} 0 \\ \\ \\ \end{bmatrix}$$

Simon's algorithm – example

$$|\phi_4\rangle = \frac{1}{\sqrt{8}}\left[|0000\rangle - |0010\rangle - |0100\rangle + |0110\rangle + |1001\rangle - |1011\rangle - |1101\rangle + |1111\rangle\right]$$

The result of the final measurement, $|y\rangle$ will be one of these eight values and each of them should satisfy the linear equation $a \cdot y = 0$

Since we know that $a = |1001\rangle$ for this example, we can check this identity

$$|1001\rangle \cdot |0000\rangle = 1 \cdot 0 + 0 \cdot 0 + 0 \cdot 0 + 1 \cdot 0 = 0$$

$$|1001\rangle \cdot |1001\rangle = 1 \cdot 1 + 0 \cdot 0 + 0 \cdot 0 + 1 \cdot 1 = 2 = 0$$

and the other 6 have the same properties

It is now necessary to collect $n - 1 = 3$ independent values of $|y\rangle$ to solve for a

Trial	$\|y\rangle$	Indep.?
1	$\|0000\rangle$	No
1	$\|0010\rangle$	Yes
1	$\|0100\rangle$	Yes
1	$\|0110\rangle$	No
1	$\|1001\rangle$	Yes

Create a matrix from the $y \cdot a = 0$ equation and the three independent values obtained

$$\begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ & & & \end{bmatrix} \begin{bmatrix} a_3 \\ a_2 \\ \\ \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \\ \end{bmatrix}$$

Simon's algorithm – example

$$|\phi_4\rangle = \frac{1}{\sqrt{8}}\left[|0000\rangle - |0010\rangle - |0100\rangle + |0110\rangle + |1001\rangle - |1011\rangle - |1101\rangle + |1111\rangle\right]$$

The result of the final measurement, $|y\rangle$ will be one of these eight values and each of them should satisfy the linear equation $a \cdot y = 0$

Since we know that $a = |1001\rangle$ for this example, we can check this identity

$$|1001\rangle \cdot |0000\rangle = 1\cdot 0 + 0\cdot 0 + 0\cdot 0 + 1\cdot 0 = 0$$

$$|1001\rangle \cdot |1001\rangle = 1\cdot 1 + 0\cdot 0 + 0\cdot 0 + 1\cdot 1 = 2 = 0$$

and the other 6 have the same properties

It is now necessary to collect $n-1=3$ independent values of $|y\rangle$ to solve for a

Trial	$\|y\rangle$	Indep.?
1	$\|0000\rangle$	No
1	$\|0010\rangle$	Yes
1	$\|0100\rangle$	Yes
1	$\|0110\rangle$	No
1	$\|1001\rangle$	Yes

Create a matrix from the $y \cdot a = 0$ equation and the three independent values obtained

$$\begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ & & & \end{bmatrix} \begin{bmatrix} a_3 \\ a_2 \\ a_1 \\ \\ \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \\ \end{bmatrix}$$

Simon's algorithm – example

$$|\phi_4\rangle = \frac{1}{\sqrt{8}}\left[|0000\rangle - |0010\rangle - |0100\rangle + |0110\rangle + |1001\rangle - |1011\rangle - |1101\rangle + |1111\rangle\right]$$

The result of the final measurement, $|y\rangle$ will be one of these eight values and each of them should satisfy the linear equation $a \cdot y = 0$

Since we know that $a = |1001\rangle$ for this example, we can check this identity

and the other 6 have the same properties

$$|1001\rangle \cdot |0000\rangle = 1\cdot 0 + 0\cdot 0 + 0\cdot 0 + 1\cdot 0 = 0$$

$$|1001\rangle \cdot |1001\rangle = 1\cdot 1 + 0\cdot 0 + 0\cdot 0 + 1\cdot 1 = 2 = 0$$

It is now necessary to collect $n - 1 = 3$ independent values of $|y\rangle$ to solve for a

Trial	$\vert y\rangle$	Indep.?
1	$\vert 0000\rangle$	No
1	$\vert 0010\rangle$	Yes
1	$\vert 0100\rangle$	Yes
1	$\vert 0110\rangle$	No
1	$\vert 1001\rangle$	Yes

Create a matrix from the $y \cdot a = 0$ equation and the three independent values obtained

$$\begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} a_3 \\ a_2 \\ a_1 \\ a_0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Simon's algorithm – example

Solve this matrix equation by Gaussian elimination

$$\begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} a_3 \\ a_2 \\ a_1 \\ a_0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Simon's algorithm – example

Solve this matrix equation by Gaussian elimination

$$\begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} a_3 \\ a_2 \\ a_1 \\ a_0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Convert the matrix to an upper triangular form by swapping rows 1 and 3

Simon's algorithm – example

Solve this matrix equation by Gaussian elimination

$$\begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} a_3 \\ a_2 \\ a_1 \\ a_0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Convert the matrix to an upper triangular form by swapping rows 1 and 3

$$\begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} a_3 \\ a_2 \\ a_1 \\ a_0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Simon's algorithm – example

Solve this matrix equation by Gaussian elimination

$$\begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} a_3 \\ a_2 \\ a_1 \\ a_0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Convert the matrix to an upper triangular form by swapping rows 1 and 3

$$\begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} a_3 \\ a_2 \\ a_1 \\ a_0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Since the bottom row of the matrix is all zeros, a_0 can be either 0 or 1

Simon's algorithm – example

Solve this matrix equation by Gaussian elimination

$$\begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} a_3 \\ a_2 \\ a_1 \\ a_0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Convert the matrix to an upper triangular form by swapping rows 1 and 3

$$\begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} a_3 \\ a_2 \\ a_1 \\ a_0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Since the bottom row of the matrix is all zeros, a_0 can be either 0 or 1

$$a_0 = 0$$

$$a_0 = 1$$

Simon's algorithm – example

Solve this matrix equation by Gaussian elimination

$$\begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} a_3 \\ a_2 \\ a_1 \\ a_0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Convert the matrix to an upper triangular form by swapping rows 1 and 3

$$\begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} a_3 \\ a_2 \\ a_1 \\ a_0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Since the bottom row of the matrix is all zeros, a_0 can be either 0 or 1

$$a_0 = 0$$

$$a_1 = 0,$$

$$a_0 = 1$$

$$a_1 = 0,$$

Simon's algorithm – example

Solve this matrix equation by Gaussian elimination

$$\begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} a_3 \\ a_2 \\ a_1 \\ a_0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Convert the matrix to an upper triangular form by swapping rows 1 and 3

$$\begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} a_3 \\ a_2 \\ a_1 \\ a_0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Since the bottom row of the matrix is all zeros, a_0 can be either 0 or 1

$$a_0 = 0$$
$$a_1 = 0, \quad a_2 = 0,$$

$$a_0 = 1$$
$$a_1 = 0, \quad a_2 = 0,$$

Simon's algorithm – example

Solve this matrix equation by Gaussian elimination

$$\begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} a_3 \\ a_2 \\ a_1 \\ a_0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Convert the matrix to an upper triangular form by swapping rows 1 and 3

$$\begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} a_3 \\ a_2 \\ a_1 \\ a_0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Since the bottom row of the matrix is all zeros, a_0 can be either 0 or 1

$$a_0 = 0$$

$$a_1 = 0, \quad a_2 = 0, \quad a_3 + a_0 = 0$$

$$a_0 = 1$$

$$a_1 = 0, \quad a_2 = 0, \quad a_3 + a_0 = 0$$

Simon's algorithm – example

Solve this matrix equation by Gaussian elimination

$$\begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} a_3 \\ a_2 \\ a_1 \\ a_0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Convert the matrix to an upper triangular form by swapping rows 1 and 3

$$\begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} a_3 \\ a_2 \\ a_1 \\ a_0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Since the bottom row of the matrix is all zeros, a_0 can be either 0 or 1

$$a_0 = 0$$

$$a_1 = 0, \quad a_2 = 0, \quad a_3 + a_0 = 0$$

$$a_3 = 0$$

$$a_0 = 1$$

$$a_1 = 0, \quad a_2 = 0, \quad a_3 + a_0 = 0$$

Simon's algorithm – example

Solve this matrix equation by Gaussian elimination

$$\begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} a_3 \\ a_2 \\ a_1 \\ a_0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Convert the matrix to an upper triangular form by swapping rows 1 and 3

$$\begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} a_3 \\ a_2 \\ a_1 \\ a_0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Since the bottom row of the matrix is all zeros, a_0 can be either 0 or 1

$$a_0 = 0$$

$$a_1 = 0, \quad a_2 = 0, \quad a_3 + a_0 = 0$$

$$a_3 = 0 \quad \longrightarrow \quad a = |0000\rangle$$

$$a_0 = 1$$

$$a_1 = 0, \quad a_2 = 0, \quad a_3 + a_0 = 0$$

Simon's algorithm – example

Solve this matrix equation by Gaussian elimination

$$\begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} a_3 \\ a_2 \\ a_1 \\ a_0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Convert the matrix to an upper triangular form by swapping rows 1 and 3

$$\begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} a_3 \\ a_2 \\ a_1 \\ a_0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Since the bottom row of the matrix is all zeros, a_0 can be either 0 or 1

$$a_0 = 0$$

$$a_1 = 0, \quad a_2 = 0, \quad a_3 + a_0 = 0$$

$$a_3 = 0 \quad \longrightarrow \quad a = |0000\rangle$$

trivial, incorrect solution

$$a_0 = 1$$

$$a_1 = 0, \quad a_2 = 0, \quad a_3 + a_0 = 0$$

Simon's algorithm – example

Solve this matrix equation by Gaussian elimination

$$\begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} a_3 \\ a_2 \\ a_1 \\ a_0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Convert the matrix to an upper triangular form by swapping rows 1 and 3

$$\begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} a_3 \\ a_2 \\ a_1 \\ a_0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Since the bottom row of the matrix is all zeros, a_0 can be either 0 or 1

$$a_0 = 0$$

$$a_1 = 0, \quad a_2 = 0, \quad a_3 + a_0 = 0$$

$$a_3 = 0 \quad \longrightarrow \quad a = |0000\rangle$$

trivial, incorrect solution

$$a_0 = 1$$

$$a_1 = 0, \quad a_2 = 0, \quad a_3 + a_0 = 0$$

$$a_3 = -1 = 1$$

Simon's algorithm – example

Solve this matrix equation by Gaussian elimination

$$\begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} a_3 \\ a_2 \\ a_1 \\ a_0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Convert the matrix to an upper triangular form by swapping rows 1 and 3

$$\begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} a_3 \\ a_2 \\ a_1 \\ a_0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Since the bottom row of the matrix is all zeros, a_0 can be either 0 or 1

$$a_0 = 0$$
$$a_1 = 0, \quad a_2 = 0, \quad a_3 + a_0 = 0$$
$$a_3 = 0 \quad \longrightarrow \quad a = |0000\rangle$$

trivial, incorrect solution

$$a_0 = 1$$
$$a_1 = 0, \quad a_2 = 0, \quad a_3 + a_0 = 0$$
$$a_3 = -1 = 1 \quad \longrightarrow \quad a = |1001\rangle$$

Simon's algorithm – example

Solve this matrix equation by Gaussian elimination

$$\begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} a_3 \\ a_2 \\ a_1 \\ a_0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Convert the matrix to an upper triangular form by swapping rows 1 and 3

$$\begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} a_3 \\ a_2 \\ a_1 \\ a_0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Since the bottom row of the matrix is all zeros, a_0 can be either 0 or 1

$$a_0 = 0$$

$$a_1 = 0, \quad a_2 = 0, \quad a_3 + a_0 = 0$$

$$a_3 = 0 \quad \longrightarrow \quad a = |0000\rangle$$

trivial, incorrect solution

$$a_0 = 1$$

$$a_1 = 0, \quad a_2 = 0, \quad a_3 + a_0 = 0$$

$$a_3 = -1 = 1 \quad \longrightarrow \quad a = |1001\rangle$$

correct solution