QUANTUM CIRCUIS and SIMPLE QUANTUM ALGORITHMS

prof. RNDr. Jozef Gruska, DrSc.

Faculty of Informatics
Masaryk University
Brno, Czech Republic

2016

Part I

Quantum circuits and simple algorithms

PROLOGUE

REVERSIBILITY

One of very special property of quantum operations, circuits and algorithms is **reversibility**.

REVERSIBILITY

One of very special property of quantum operations, circuits and algorithms is **reversibility**.

An operation is reversible if its outputs uniquely determine its inputs.

REVERSIBILITY

One of very special property of quantum operations, circuits and algorithms is **reversibility**.

An operation is reversible if its outputs uniquely determine its inputs.

$(a, b) \rightarrow a + b$

a non-reversible operation

$(a, b) \rightarrow (a + b, a - b)$

a reversible operation

REVERSIBILITY

One of very special property of quantum operations, circuits and algorithms is **reversibility**.

An operation is reversible if its outputs uniquely determine its inputs.

$(a, b) \rightarrow a + b$

a non-reversible operation

$(a, b) \rightarrow (a + b, a - b)$

a reversible operation

$a \rightarrow f(a)$

$(a, 0) \rightarrow (a, f(a))$

REVERSIBILITY

One of very special property of quantum operations, circuits and algorithms is **reversibility**.

An operation is reversible if its outputs uniquely determine its inputs.

$(a, b) \rightarrow a + b$

a non-reversible operation

$(a, b) \rightarrow (a + b, a - b)$

a reversible operation

$a \rightarrow f(a)$

Mapping that can be reversible

$(a, 0) \rightarrow (a, f(a))$

surely reversible operation

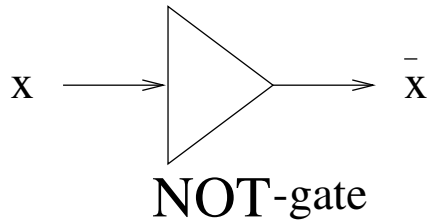
Reverisbility is of importance also for classical computing.

SET of CLASSICAL REVERSIBLE GATES

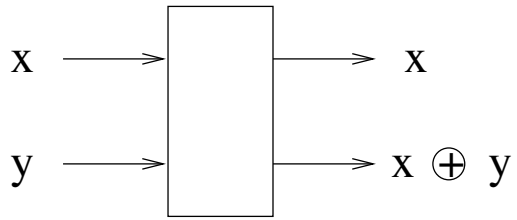
A universal set of three reversible classical gates: NOT gate, XOR or CNOT gate and Toffoli or CCNOT gate.

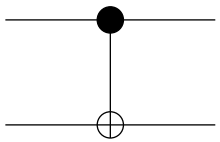
SET of CLASSICAL REVERSIBLE GATES

A universal set of three reversible classical gates: NOT gate, XOR or CNOT gate and Toffoli or CCNOT gate.

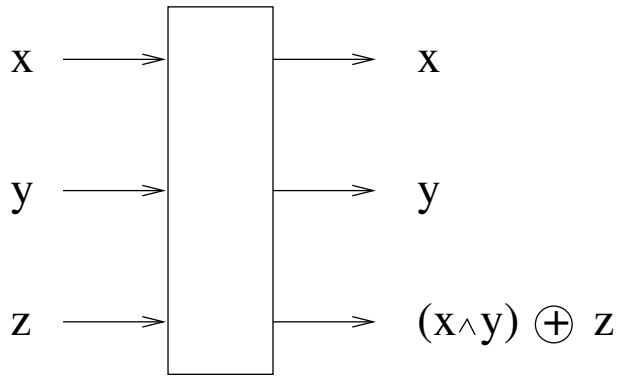


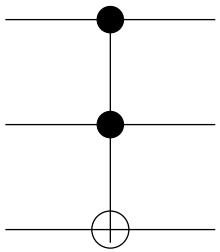
NOT-gate





CNOT = XOR gate





BRIEF HISTORY of QUANTUM COMPUTATION

BRIEF HISTORY of QUANTUM COMPUTATION

1970 Landauer demonstrated importance of reversibility for minimal energy computation;

BRIEF HISTORY of QUANTUM COMPUTATION

1970 Landauer demonstrated importance of reversibility for minimal energy computation;

1973 Bennett showed the existence of universal reversible Turing machines;

BRIEF HISTORY of QUANTUM COMPUTATION

1970 Landauer demonstrated importance of reversibility for minimal energy computation;

1973 Bennett showed the existence of universal reversible Turing machines;

1981 Toffoli-Fredkin designed a universal reversible gate for Boolean logic;

BRIEF HISTORY of QUANTUM COMPUTATION

1970 Landauer demonstrated importance of reversibility for minimal energy computation;

1973 Bennett showed the existence of universal reversible Turing machines;

1981 Toffoli-Fredkin designed a universal reversible gate for Boolean logic;

1982 Benioff showed that quantum processes are at least as powerful as Turing machines;

BRIEF HISTORY of QUANTUM COMPUTATION

1970 Landauer demonstrated importance of reversibility for minimal energy computation;

1973 Bennett showed the existence of universal reversible Turing machines;

1981 Toffoli-Fredkin designed a universal reversible gate for Boolean logic;

1982 Benioff showed that quantum processes are at least as powerful as Turing machines;

1982 Feynman demonstrated that quantum physics cannot be simulated effectively on classical computers;

BRIEF HISTORY of QUANTUM COMPUTATION

1970 Landauer demonstrated importance of reversibility for minimal energy computation;

1973 Bennett showed the existence of universal reversible Turing machines;

1981 Toffoli-Fredkin designed a universal reversible gate for Boolean logic;

1982 Benioff showed that quantum processes are at least as powerful as Turing machines;

1982 Feynman demonstrated that quantum physics cannot be simulated effectively on classical computers;

1984 Quantum cryptographic protocol BB84 was published, by Bennett and Brassard, for absolutely secure generation of shared secret random classical keys.

BRIEF HISTORY of QUANTUM COMPUTATION

1970 Landauer demonstrated importance of reversibility for minimal energy computation;

1973 Bennett showed the existence of universal reversible Turing machines;

1981 Toffoli-Fredkin designed a universal reversible gate for Boolean logic;

1982 Benioff showed that quantum processes are at least as powerful as Turing machines;

1982 Feynman demonstrated that quantum physics cannot be simulated effectively on classical computers;

1984 Quantum cryptographic protocol BB84 was published, by Bennett and Brassard, for absolutely secure generation of shared secret random classical keys.

1985 Deutsch showed the existence of a universal quantum Turing machine.

BRIEF HISTORY of QUANTUM COMPUTATION

1970 Landauer demonstrated importance of reversibility for minimal energy computation;

1973 Bennett showed the existence of universal reversible Turing machines;

1981 Toffoli-Fredkin designed a universal reversible gate for Boolean logic;

1982 Benioff showed that quantum processes are at least as powerful as Turing machines;

1982 Feynman demonstrated that quantum physics cannot be simulated effectively on classical computers;

1984 Quantum cryptographic protocol BB84 was published, by Bennett and Brassard, for absolutely secure generation of shared secret random classical keys.

1985 Deutsch showed the existence of a universal quantum Turing machine.

1989 First cryptographic experiment for transmission of photons, for distance 32.5cm was performed by Bennett, Brassard and Smolin.

BRIEF HISTORY of QUANTUM COMPUTATION

1970 Landauer demonstrated importance of reversibility for minimal energy computation;

1973 Bennett showed the existence of universal reversible Turing machines;

1981 Toffoli-Fredkin designed a universal reversible gate for Boolean logic;

1982 Benioff showed that quantum processes are at least as powerful as Turing machines;

1982 Feynman demonstrated that quantum physics cannot be simulated effectively on classical computers;

1984 Quantum cryptographic protocol BB84 was published, by Bennett and Brassard, for absolutely secure generation of shared secret random classical keys.

1985 Deutsch showed the existence of a universal quantum Turing machine.

1989 First cryptographic experiment for transmission of photons, for distance 32.5cm was performed by Bennett, Brassard and Smolin.

1993 Bernstein-Vazirani-Yao showed the existence of an efficient universal quantum Turing machine;

1993 Quantum teleportation was discovered, by Bennett et al.

1993 Quantum teleportation was discovered, by Bennett et al.

1994 Shor discovered a polynomial time quantum algorithm for factorization;

1993 Quantum teleportation was discovered, by Bennett et al.

1994 Shor discovered a polynomial time quantum algorithm for factorization;

Cryptographic experiments were performed for the distance of 10km (using fibers).

1993 Quantum teleportation was discovered, by Bennett et al.

1994 Shor discovered a polynomial time quantum algorithm for factorization;

Cryptographic experiments were performed for the distance of 10km (using fibers).

1994 Quantum cryptography went through an experimental stage;

1993 Quantum teleportation was discovered, by Bennett et al.

1994 Shor discovered a polynomial time quantum algorithm for factorization;

Cryptographic experiments were performed for the distance of 10km (using fibers).

1994 Quantum cryptography went through an experimental stage;

1995 DiVincenzo designed a universal gate with two inputs and outputs;

1993 Quantum teleportation was discovered, by Bennett et al.

1994 Shor discovered a polynomial time quantum algorithm for factorization;

Cryptographic experiments were performed for the distance of 10km (using fibers).

1994 Quantum cryptography went through an experimental stage;

1995 DiVincenzo designed a universal gate with two inputs and outputs;

1995 Cirac and Zoller demonstrated a chance to build quantum computers using existing technologies.

1993 Quantum teleportation was discovered, by Bennett et al.

1994 Shor discovered a polynomial time quantum algorithm for factorization;

Cryptographic experiments were performed for the distance of 10km (using fibers).

1994 Quantum cryptography went through an experimental stage;

1995 DiVincenzo designed a universal gate with two inputs and outputs;

1995 Cirac and Zoller demonstrated a chance to build quantum computers using existing technologies.

1995 Shor showed the existence of quantum error-correcting codes.

1993 Quantum teleportation was discovered, by Bennett et al.

1994 Shor discovered a polynomial time quantum algorithm for factorization;

Cryptographic experiments were performed for the distance of 10km (using fibers).

1994 Quantum cryptography went through an experimental stage;

1995 DiVincenzo designed a universal gate with two inputs and outputs;

1995 Cirac and Zoller demonstrated a chance to build quantum computers using existing technologies.

1995 Shor showed the existence of quantum error-correcting codes.

1996 The existence of quantum fault-tolerant computation was shown by Shor.

QUANTUM SYSTEM versus QUANTUM SPACE

Hilbert space H_n is n-dimensional complex vector space with

QUANTUM SYSTEM versus QUANTUM SPACE

Hilbert space H_n is n-dimensional complex vector space with

scalar product

$$\langle\psi|\phi\rangle = \sum_{i=1}^{n} \phi_i \psi_i^* \quad \text{of vectors } |\phi\rangle = \begin{vmatrix} \phi_1 \\ \phi_2 \\ \vdots \\ \phi_n \end{vmatrix}, |\psi\rangle = \begin{vmatrix} \psi_1 \\ \psi_2 \\ \vdots \\ \psi_n \end{vmatrix},$$

QUANTUM SYSTEM versus QUANTUM SPACE

Hilbert space H_n is n-dimensional complex vector space with

scalar product

$$\langle\psi|\phi\rangle = \sum_{i=1}^{n} \phi_i \psi_i^* \quad \text{of vectors } |\phi\rangle = \begin{vmatrix} \phi_1 \\ \phi_2 \\ \vdots \\ \phi_n \end{vmatrix}, |\psi\rangle = \begin{vmatrix} \psi_1 \\ \psi_2 \\ \vdots \\ \psi_n \end{vmatrix},$$

norm of vectors

$$||\phi|| = \sqrt{|\langle\phi|\phi\rangle|}$$

and the **metric**

QUANTUM SYSTEM versus QUANTUM SPACE

Hilbert space H_n is n-dimensional complex vector space with

scalar product

$$\langle\psi|\phi\rangle = \sum_{i=1}^{n} \phi_i \psi_i^* \quad \text{of vectors } |\phi\rangle = \begin{vmatrix} \phi_1 \\ \phi_2 \\ \vdots \\ \phi_n \end{vmatrix}, |\psi\rangle = \begin{vmatrix} \psi_1 \\ \psi_2 \\ \vdots \\ \psi_n \end{vmatrix},$$

norm of vectors

$$||\phi|| = \sqrt{|\langle\phi|\phi\rangle|}$$

and the **metric**

$$\text{dist}(\phi, \psi) = ||\phi - \psi||.$$

This allows us to introduce on $\mathcal{H}$ a topology and such concepts as continuity.

QUANTUM SYSTEM versus QUANTUM SPACE

Hilbert space H_n is n-dimensional complex vector space with

scalar product

$$\langle\psi|\phi\rangle = \sum_{i=1}^{n} \phi_i \psi_i^* \quad \text{of vectors } |\phi\rangle = \begin{vmatrix} \phi_1 \\ \phi_2 \\ \vdots \\ \phi_n \end{vmatrix}, |\psi\rangle = \begin{vmatrix} \psi_1 \\ \psi_2 \\ \vdots \\ \psi_n \end{vmatrix},$$

norm of vectors

$$||\phi|| = \sqrt{|\langle\phi|\phi\rangle|}$$

and the **metric**

$$\text{dist}(\phi, \psi) = ||\phi - \psi||.$$

This allows us to introduce on $\mathcal{H}$ a topology and such concepts as continuity. Elements (vectors) of a Hilbert space $\mathcal{H}$ are usually called **pure states** of H.

ORTHOGONALITY of PURE STATES

ORTHOGONALITY of PURE STATES

Two quantum states $|\phi\rangle$ and $|\psi\rangle$ are called orthogonal if their scalar product is zero, that is if

$$\langle\phi|\psi\rangle = 0.$$

ORTHOGONALITY of PURE STATES

Two quantum states $|\phi\rangle$ and $|\psi\rangle$ are called orthogonal if their scalar product is zero, that is if

$$\langle\phi|\psi\rangle = 0.$$

Two pure quantum states are physically perfectly distinguishable only if they are orthogonal.

ORTHOGONALITY of PURE STATES

Two quantum states $|\phi\rangle$ and $|\psi\rangle$ are called orthogonal if their scalar product is zero, that is if

$$\langle\phi|\psi\rangle = 0.$$

Two pure quantum states are physically perfectly distinguishable only if they are orthogonal.

In every Hilbert space there are so-called orthogonal bases all states of which are mutually orthogonal.

QUBITS

QUBITS

A **qubit** - a two-level quantum system is a quantum state in H_2

$$|\phi\rangle = \alpha|0\rangle + \beta|1\rangle$$

QUBITS

A **qubit** - a two-level quantum system is a quantum state in H_2

$$|\phi\rangle = \alpha|0\rangle + \beta|1\rangle$$

where $\alpha, \beta \in \mathbf{C}$ are such that $|\alpha|^2 + |\beta|^2 = 1$ and

QUBITS

A **qubit** - a two-level quantum system is a quantum state in H_2

$$|\phi\rangle = \alpha|0\rangle + \beta|1\rangle$$

where $\alpha, \beta \in \mathbf{C}$ are such that $|\alpha|^2 + |\beta|^2 = 1$ and

$\{|0\rangle, |1\rangle\}$ is a (**standard**) **basis** of H_2

QUBITS

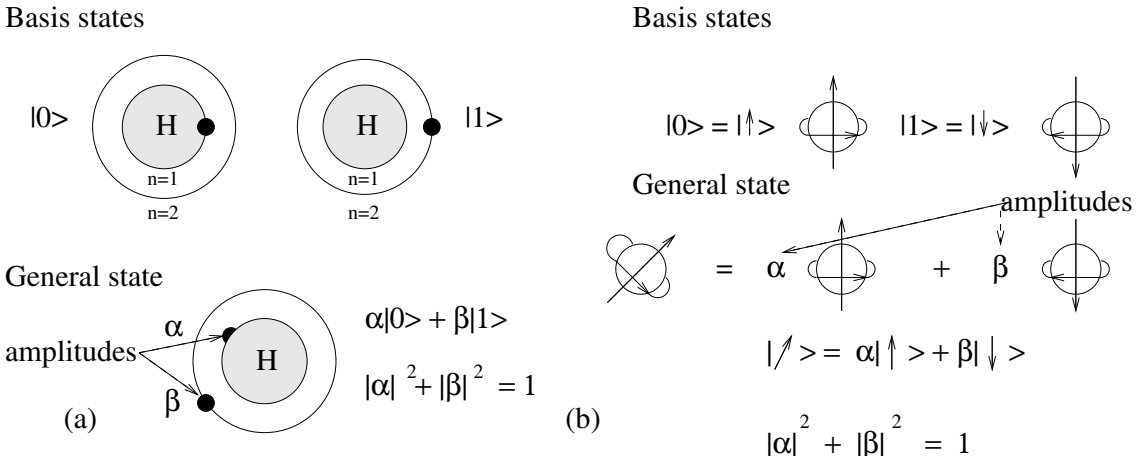
A **qubit** - a two-level quantum system is a quantum state in H_2

$$|\phi\rangle = \alpha|0\rangle + \beta|1\rangle$$

where $\alpha, \beta \in \mathbf{C}$ are such that $|\alpha|^2 + |\beta|^2 = 1$ and

$$\{|0\rangle, |1\rangle\} \quad \text{is a (\textbf{standard}) \textbf{basis} of} \quad H_2$$

EXAMPLE: Representation of qubits by (a) an electron in a Hydrogen atom; (b) a spin-$\frac{1}{2}$ particle:



QUANTUM REGISTERS

QUANTUM REGISTERS

Any ordered sequence of n quantum qubit systems creates so-called quantum n-qubit register.

QUANTUM REGISTERS

Any ordered sequence of n quantum qubit systems creates so-called quantum n-qubit register.

Hilbert space corresponding to an n-qubit register is n-fold tensor product of two-dimensional Hilbert spaces

$$\mathcal{H}_{2^n} = \bigotimes_{i=1}^{n} \mathcal{H}_2.$$

QUANTUM REGISTERS

Any ordered sequence of n quantum qubit systems creates so-called quantum n-qubit register.

Hilbert space corresponding to an n-qubit register is n-fold tensor product of two-dimensional Hilbert spaces

$$\mathcal{H}_{2^n} = \bigotimes_{i=1}^{n} \mathcal{H}_2.$$

Since vectors $|0\rangle$ and $|1\rangle$ form a basis of H_2, one of the basis of $\mathcal{H}_{2^n}$, so-called computational basis, consists of all possible n-fold tensor products where $b_i \in \{0, 1\}$ for all i.

$$|b_1\rangle \otimes |b_2\rangle \otimes \ldots \otimes |b_n\rangle = |b_1 b_2 \ldots b_n\rangle.$$

QUANTUM REGISTERS

Any ordered sequence of n quantum qubit systems creates so-called quantum n-qubit register.

Hilbert space corresponding to an n-qubit register is n-fold tensor product of two-dimensional Hilbert spaces

$$\mathcal{H}_{2^n} = \bigotimes_{i=1}^{n} \mathcal{H}_2.$$

Since vectors $|0\rangle$ and $|1\rangle$ form a basis of H_2, one of the basis of $\mathcal{H}_{2^n}$, so-called computational basis, consists of all possible n-fold tensor products where $b_i \in \{0, 1\}$ for all i.

$$|b_1\rangle \otimes |b_2\rangle \otimes \ldots \otimes |b_n\rangle = |b_1 b_2 \ldots b_n\rangle.$$

Example A two-qubit register has as a computational basis vectors

$$|00\rangle = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} \quad |01\rangle = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} \quad |10\rangle = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} \quad |11\rangle = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

QUANTUM STATES and PROJECTION MEASUREMENTS

In case an orthonormal basis $\{\beta_i\}_{i=1}^n$ is chosen in $\mathcal{H}_n$, any state $|\phi\rangle \in \mathcal{H}_n$ can be expressed in the form

$$|\phi\rangle = \sum_{i=1}^{n} a_i |\beta_i\rangle, \quad \sum_{i=1}^{n} |a_i|^2 = 1,$$

QUANTUM STATES and PROJECTION MEASUREMENTS

In case an orthonormal basis $\{\beta_i\}_{i=1}^n$ is chosen in $\mathcal{H}_n$, any state $|\phi\rangle \in \mathcal{H}_n$ can be expressed in the form

$$|\phi\rangle = \sum_{i=1}^{n} a_i |\beta_i\rangle, \quad \sum_{i=1}^{n} |a_i|^2 = 1,$$

where

$a_i = \langle\beta_i|\phi\rangle$ are called probability amplitudes

QUANTUM STATES and PROJECTION MEASUREMENTS

In case an orthonormal basis $\{\beta_i\}_{i=1}^n$ is chosen in $\mathcal{H}_n$, any state $|\phi\rangle \in \mathcal{H}_n$ can be expressed in the form

$$|\phi\rangle = \sum_{i=1}^{n} a_i |\beta_i\rangle, \quad \sum_{i=1}^{n} |a_i|^2 = 1,$$

where

$a_i = \langle\beta_i|\phi\rangle$ are called probability amplitudes

and

their squares, $|a_i|^2 = \langle\phi\beta_i\rangle\langle\beta_i|\phi\rangle$, provide probabilities

that if the state $|\phi\rangle$ is measured with respect to the basis $\{\beta_i\}_{i=1}^n$,

QUANTUM STATES and PROJECTION MEASUREMENTS

In case an orthonormal basis $\{\beta_i\}_{i=1}^n$ is chosen in $\mathcal{H}_n$, any state $|\phi\rangle \in \mathcal{H}_n$ can be expressed in the form

$$|\phi\rangle = \sum_{i=1}^{n} a_i |\beta_i\rangle, \quad \sum_{i=1}^{n} |a_i|^2 = 1,$$

where

$a_i = \langle\beta_i|\phi\rangle$ are called probability amplitudes

and

their squares, $|a_i|^2 = \langle\phi\beta_i\rangle\langle\beta_i|\phi\rangle$, provide probabilities

that if the state $|\phi\rangle$ is measured with respect to the basis $\{\beta_i\}_{i=1}^n$, then the state $|\phi\rangle$ collapses into the state $|\beta_i\rangle$ with probability $|a_i|^2$.

QUANTUM STATES and PROJECTION MEASUREMENTS

In case an orthonormal basis $\{\beta_i\}_{i=1}^n$ is chosen in $\mathcal{H}_n$, any state $|\phi\rangle \in \mathcal{H}_n$ can be expressed in the form

$$|\phi\rangle = \sum_{i=1}^{n} a_i |\beta_i\rangle, \quad \sum_{i=1}^{n} |a_i|^2 = 1,$$

where

$a_i = \langle\beta_i|\phi\rangle$ are called probability amplitudes

and

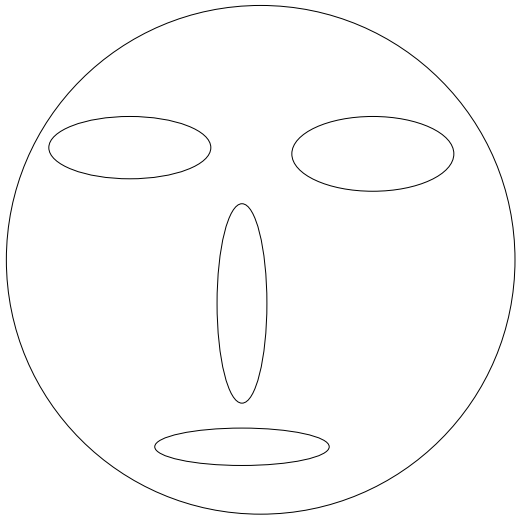
their squares, $|a_i|^2 = \langle\phi\beta_i\rangle\langle\beta_i|\phi\rangle$, provide probabilities

that if the state $|\phi\rangle$ is measured with respect to the basis $\{\beta_i\}_{i=1}^n$, then the state $|\phi\rangle$ collapses into the state $|\beta_i\rangle$ with probability $|a_i|^2$.

The classical "outcome" of a projection (von Neumann) measurement of the state $|\phi\rangle$ with respect to the basis $\{\beta_i\}_{i=1}^n$ is the index i of that state $|\beta_i\rangle$ into which the state $|\phi\rangle$ collapses.

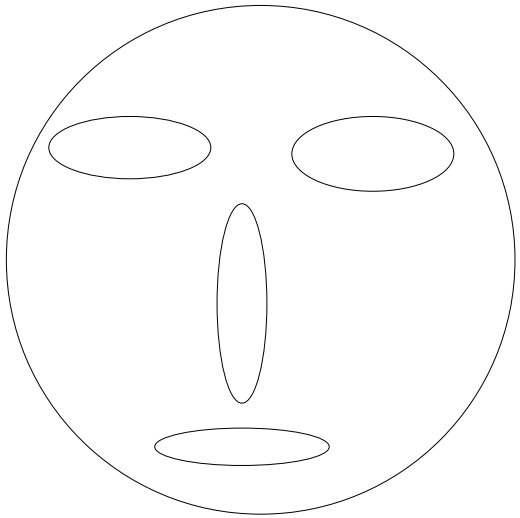
QUANTUM MEASUREMENT - II

A quantum state is observed (measured) with respect to an **observable** — a decomposition of a given Hilbert space into orthogonal subspaces (such that each vector can be uniquely represented as a sum of vectors of these subspaces).



QUANTUM MEASUREMENT - II

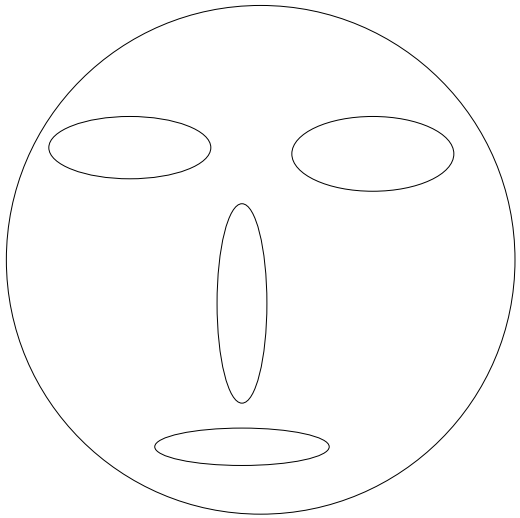
A quantum state is observed (measured) with respect to an **observable** — a decomposition of a given Hilbert space into orthogonal subspaces (such that each vector can be uniquely represented as a sum of vectors of these subspaces).



There are two outcomes of a projection measurement of a state $|\phi\rangle$:

QUANTUM MEASUREMENT - II

A quantum state is observed (measured) with respect to an **observable** — a decomposition of a given Hilbert space into orthogonal subspaces (such that each vector can be uniquely represented as a sum of vectors of these subspaces).

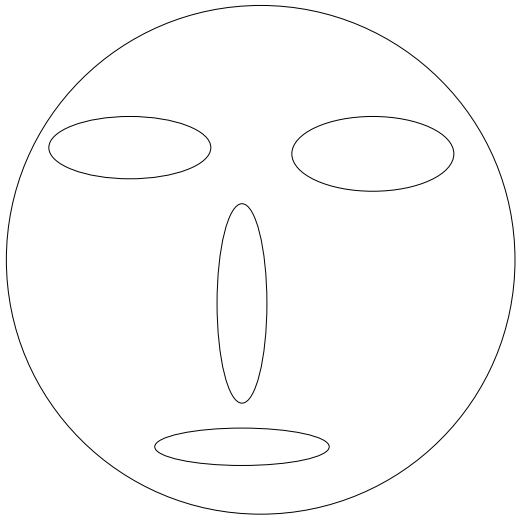


There are two outcomes of a projection measurement of a state $|\phi\rangle$:

1. Classical information into which subspace projection of $|\phi\rangle$ was made.

QUANTUM MEASUREMENT - II

A quantum state is observed (measured) with respect to an **observable** — a decomposition of a given Hilbert space into orthogonal subspaces (such that each vector can be uniquely represented as a sum of vectors of these subspaces).

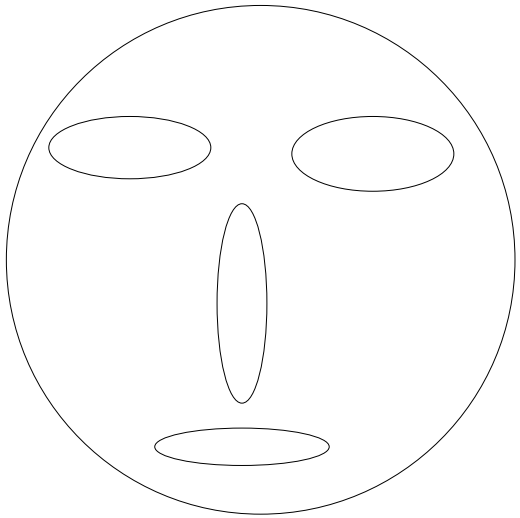


There are two outcomes of a projection measurement of a state $|\phi\rangle$:

1. Classical information into which subspace projection of $|\phi\rangle$ was made.
2. A new quantum state $|\phi'\rangle$ into which the state $|\phi\rangle$ collapses.

QUANTUM MEASUREMENT - II

A quantum state is observed (measured) with respect to an **observable** — a decomposition of a given Hilbert space into orthogonal subspaces (such that each vector can be uniquely represented as a sum of vectors of these subspaces).



There are two outcomes of a projection measurement of a state $|\phi\rangle$:

1. Classical information into which subspace projection of $|\phi\rangle$ was made.
2. A new quantum state $|\phi'\rangle$ into which the state $|\phi\rangle$ collapses.

The subspace into which projection is made is chosen **randomly** and the corresponding probability is uniquely determined by the amplitudes at the representation of $|\phi\rangle$ at the basis states of the subspace.

QUANTUM GATES and CIRCUITS

- Quantum gates are fundamental primitives for quantum computations.

QUANTUM GATES and CIRCUITS

- Quantum gates are fundamental primitives for quantum computations.
- Quantum circuits are the simplest and best model to express quantum algorithms.

MAIN MODELS of CLASSICAL PROCESSORS - I.

Main classical models of processors are:

- Finite automata (deterministic, non-deterministic, probabilistic, ultrametric,...., one-way, two-way,...; one-tape, multi-tape,....)

MAIN MODELS of CLASSICAL PROCESSORS - I.

Main classical models of processors are:

- Finite automata (deterministic, non-deterministic, probabilistic, ultrametric,...., one-way, two-way,...; one-tape, multi-tape,....)
- Turing machines (with one or more tapes, with one or more heads, with one or more dimensional tapes - deterministic, non-deterministic, probabilistic,...) - models of universal processors

MAIN MODELS of CLASSICAL PROCESSORS - I.

Main classical models of processors are:

- Finite automata (deterministic, non-deterministic, probabilistic, ultrametric,...., one-way, two-way,...; one-tape, multi-tape,....)
- Turing machines (with one or more tapes, with one or more heads, with one or more dimensional tapes - deterministic, non-deterministic, probabilistic,...) - models of universal processors
- Uniform classes of circuits - models of universal processors

MAIN MODELS of CLASSICAL PROCESSORS - I.

Main classical models of processors are:

- Finite automata (deterministic, non-deterministic, probabilistic, ultrametric,...., one-way, two-way,...; one-tape, multi-tape,....)
- Turing machines (with one or more tapes, with one or more heads, with one or more dimensional tapes - deterministic, non-deterministic, probabilistic,...) - models of universal processors
- Uniform classes of circuits - models of universal processors
- Cellular automata (one-, two-, three- and more dimensional)– models of universal processors.

MAIN MODELS of CLASSICAL PROCESSORS - I.

Main classical models of processors are:

- Finite automata (deterministic, non-deterministic, probabilistic, ultrametric,...., one-way, two-way,...; one-tape, multi-tape,....)
- Turing machines (with one or more tapes, with one or more heads, with one or more dimensional tapes - deterministic, non-deterministic, probabilistic,...) - models of universal processors
- Uniform classes of circuits - models of universal processors
- Cellular automata (one-, two-, three- and more dimensional)– models of universal processors.

Of importance, especially for an understanding of the power and development of methods (of programming) for very powerful real processors are also the following models:

- RAM - Random access machines
- PRAM - Parallel and shared memory random access machines

MAIN MODELS of CLASSICAL PROCESSORS - I.

Main classical models of processors are:

- Finite automata (deterministic, non-deterministic, probabilistic, ultrametric,...., one-way, two-way,...; one-tape, multi-tape,....)
- Turing machines (with one or more tapes, with one or more heads, with one or more dimensional tapes - deterministic, non-deterministic, probabilistic,...) - models of universal processors
- Uniform classes of circuits - models of universal processors
- Cellular automata (one-, two-, three- and more dimensional)– models of universal processors.

Of importance, especially for an understanding of the power and development of methods (of programming) for very powerful real processors are also the following models:

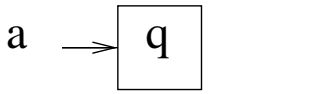
- RAM - Random access machines
- PRAM - Parallel and shared memory random access machines

and also

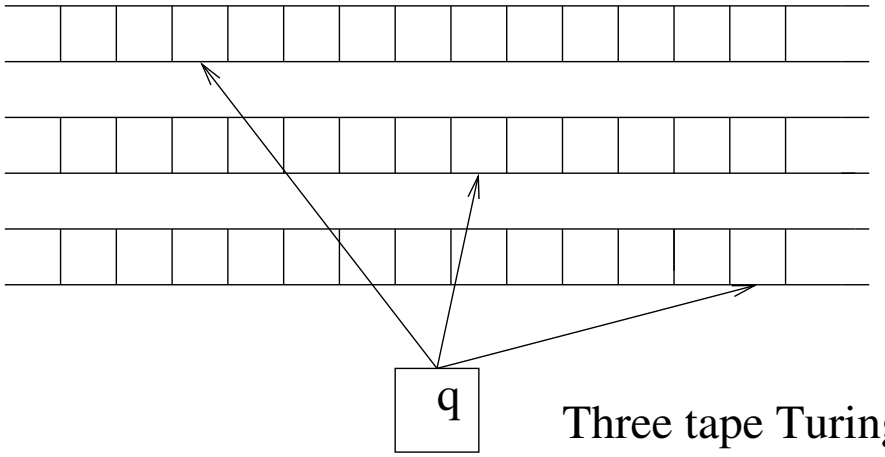
- Interactive proof systems – to model computations by interactions.

MAIN MODELS of CLASSICAL PROCESSORS - II.

MAIN MODELS of CLASSICAL PROCESSORS - II.

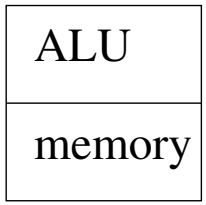


Finite automaton

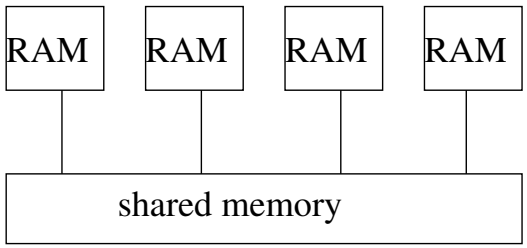


Three tape Turing machine

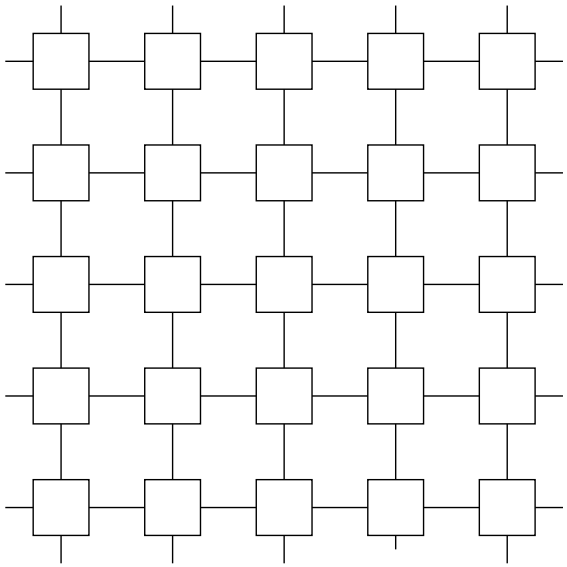
RAM



Operations: Load, Store
Add, Subtract
Jump, Jump–if



PRAM



Two–dimensional cellular automaton

MAIN MODELS of QUANTUM PROCESSORS

MAIN MODELS of QUANTUM PROCESSORS

- Unitary operations based quantum circuits

MAIN MODELS of QUANTUM PROCESSORS

- Unitary operations based quantum circuits
- Unitary operations based quantum finite automata

MAIN MODELS of QUANTUM PROCESSORS

- Unitary operations based quantum circuits
- Unitary operations based quantum finite automata
- Unitary operations based Turing machines

MAIN MODELS of QUANTUM PROCESSORS

- Unitary operations based quantum circuits
- Unitary operations based quantum finite automata
- Unitary operations based Turing machines
- Unitary operations based cellular automata

MAIN MODELS of QUANTUM PROCESSORS

- Unitary operations based quantum circuits
- Unitary operations based quantum finite automata
- Unitary operations based Turing machines
- Unitary operations based cellular automata
- Measurements based quantum circuits

MAIN MODELS of QUANTUM PROCESSORS

- Unitary operations based quantum circuits
- Unitary operations based quantum finite automata
- Unitary operations based Turing machines
- Unitary operations based cellular automata
- Measurements based quantum circuits
- Measurements based quantum Turing machines

MAIN MODELS of QUANTUM PROCESSORS

- Unitary operations based quantum circuits
- Unitary operations based quantum finite automata
- Unitary operations based Turing machines
- Unitary operations based cellular automata
- Measurements based quantum circuits
- Measurements based quantum Turing machines

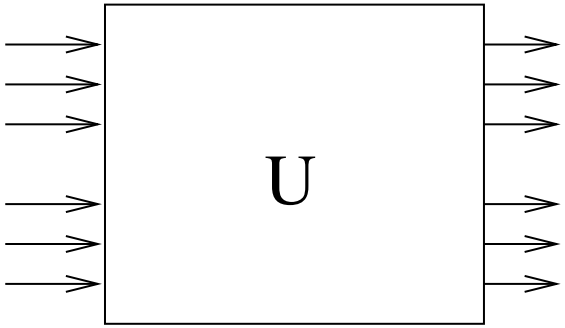
- Emerging idea: Classically controlled quantum computation (automata).

QUANTUM GATES

Unitarity is the main new requirement quantum gates have to satisfy.

QUANTUM GATES

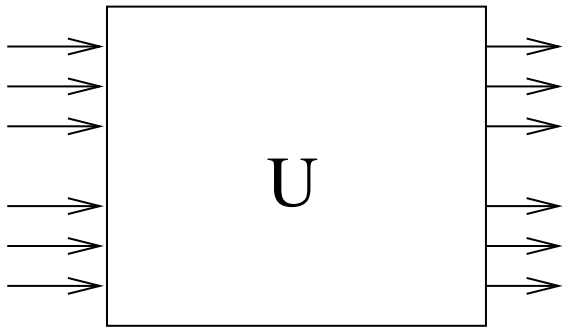
Unitarity is the main new requirement quantum gates have to satisfy.



Definition A **quantum gate** with n inputs and n outputs is specified by a unitary operator $U : \mathcal{H}_{2^n} \rightarrow \mathcal{H}_{2^n}$, and it is represented by a unitary matrix A_U of degree 2^n.

QUANTUM GATES

Unitarity is the main new requirement quantum gates have to satisfy.



Definition A **quantum gate** with n inputs and n outputs is specified by a unitary operator $U : \mathcal{H}_{2^n} \rightarrow \mathcal{H}_{2^n}$, and it is represented by a unitary matrix A_U of degree 2^n.

Example: The so-called Hadamard gates are represented by matrices

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

UNITARY MATRICES

UNITARY MATRICES

A matrix A is **unitary** *if for A and its adjoin matrix $A^\dagger$ (with $A^\dagger_{ij} = (A_{ji})^*$) it holds:*

$$A \cdot A^\dagger = A^\dagger \cdot A = I$$

UNITARY MATRICES

A matrix A is **unitary** *if for A and its adjoin matrix* $A^\dagger$ *(with* $A^\dagger_{ij} = (A_{ji})^*$*) it holds:*

$$A \cdot A^\dagger = A^\dagger \cdot A = I$$

Another view of unitarity (for mappings): unitary mapping U is a linear mapping that preserves the inner product, that is

$$\langle U\phi | U\psi \rangle = \langle \phi | \psi \rangle.$$

BRA-KET NOTATION

BRA-KET NOTATION

Dirac introduced a very handy notation, so called **bra-ket notation**, to deal with amplitudes, quantum states and linear functionals $f : H \to \mathbf{C}$.

BRA-KET NOTATION

Dirac introduced a very handy notation, so called **bra-ket notation**, to deal with amplitudes, quantum states and linear functionals $f : H \to \mathbf{C}$.

If $\psi, \phi \in H$, then

$\langle\psi|\phi\rangle$ — a number - a scalar product of ψ and ϕ (an amplitude of going from ϕ to ψ).

BRA-KET NOTATION

Dirac introduced a very handy notation, so called **bra-ket notation**, to deal with amplitudes, quantum states and linear functionals $f : H \to \mathbf{C}$.

If $\psi, \phi \in H$, then

$\langle\psi|\phi\rangle$ — a number - a scalar product of ψ and ϕ (an amplitude of going from ϕ to ψ).

$|\phi\rangle$ — **ket-vector** — a column vector - an equivalent to ϕ

BRA-KET NOTATION

Dirac introduced a very handy notation, so called **bra-ket notation**, to deal with amplitudes, quantum states and linear functionals $f : H \to \mathbf{C}$.

If $\psi, \phi \in H$, then

$\langle\psi|\phi\rangle$ — a number - a scalar product of ψ and ϕ (an amplitude of going from ϕ to ψ).

$|\phi\rangle$ — **ket-vector** — a column vector - an equivalent to ϕ

$\langle\psi|$ — **bra-vector** – a row vector - the conjugate transpose of $|\psi\rangle$ – a linear functional on H such that $\langle\psi|(|\phi\rangle) = \langle\psi|\phi\rangle$

BRA-KET NOTATION

Dirac introduced a very handy notation, so called **bra-ket notation**, to deal with amplitudes, quantum states and linear functionals $f : H \to \mathbf{C}$.

If $\psi, \phi \in H$, then

$\langle\psi|\phi\rangle$ — a number - a scalar product of ψ and ϕ (an amplitude of going from ϕ to ψ).

$|\phi\rangle$ — **ket-vector** — a column vector - an equivalent to ϕ

$\langle\psi|$ — **bra-vector** – a row vector - the conjugate transpose of $|\psi\rangle$ – a linear functional on H such that $\langle\psi|(|\phi\rangle) = \langle\psi|\phi\rangle$

Example If $\phi = (\phi_1, \ldots, \phi_n)$ and $\psi = (\psi_1, \ldots, \psi_n)$, then

$$\text{ket vector - } |\phi\rangle = \begin{pmatrix} \phi_1 \\ \vdots \\ \phi_n \end{pmatrix} \quad \text{and} \quad \langle\psi| = (\psi_1^*, \ldots, \psi_n^*) \quad - \text{bra-vector}$$

and

BRA-KET NOTATION

Dirac introduced a very handy notation, so called **bra-ket notation**, to deal with amplitudes, quantum states and linear functionals $f : H \to \mathbf{C}$.

If $\psi, \phi \in H$, then

$\langle\psi|\phi\rangle$ — a number - a scalar product of ψ and ϕ (an amplitude of going from ϕ to ψ).

$|\phi\rangle$ — **ket-vector** — a column vector - an equivalent to ϕ

$\langle\psi|$ — **bra-vector** – a row vector - the conjugate transpose of $|\psi\rangle$ – a linear functional on H such that $\langle\psi|(|\phi\rangle) = \langle\psi|\phi\rangle$

Example If $\phi = (\phi_1, \ldots, \phi_n)$ and $\psi = (\psi_1, \ldots, \psi_n)$, then

$$\text{ket vector - } |\phi\rangle = \begin{pmatrix} \phi_1 \\ \vdots \\ \phi_n \end{pmatrix} \quad \text{and} \quad \langle\psi| = (\psi_1^*, \ldots, \psi_n^*) \quad - \text{bra-vector}$$

and

$$\text{inner product - scalar product: } \langle\phi|\psi\rangle = \sum_{i=1}^{n} \phi_i^* \psi_i$$

ROTATION GATES

ROTATION GATES

Example The following one parameter set of **rotation gates** (represented by matrices) is also often used:

ROTATION GATES

Example The following one parameter set of **rotation gates** (represented by matrices) is also often used:

Rotations around axes:

ROTATION GATES

Example The following one parameter set of **rotation gates** (represented by matrices) is also often used:

Rotations around axes:

$$R_x(\theta) = \begin{pmatrix} \cos\theta & i\sin\theta \\ i\sin\theta & \cos\theta \end{pmatrix},$$

ROTATION GATES

Example The following one parameter set of **rotation gates** (represented by matrices) is also often used:

Rotations around axes:

$$R_x(\theta) = \begin{pmatrix} \cos\theta & i\sin\theta \\ i\sin\theta & \cos\theta \end{pmatrix}, \qquad R_y(\theta) = \begin{pmatrix} i\cos\theta & \sin\theta \\ \sin\theta & i\cos\theta \end{pmatrix},$$

ROTATION GATES

Example The following one parameter set of **rotation gates** (represented by matrices) is also often used:

Rotations around axes:

$$R_x(\theta) = \begin{pmatrix} \cos\theta & i\sin\theta \\ i\sin\theta & \cos\theta \end{pmatrix}, \quad R_y(\theta) = \begin{pmatrix} i\cos\theta & \sin\theta \\ \sin\theta & i\cos\theta \end{pmatrix},$$

$$R_z(\theta) = \begin{pmatrix} e^{i\theta} & 0 \\ 0 & e^{-i\theta} \end{pmatrix},$$

ROTATION GATES

Example The following one parameter set of **rotation gates** (represented by matrices) is also often used:

Rotations around axes:

$$R_x(\theta) = \begin{pmatrix} \cos\theta & i\sin\theta \\ i\sin\theta & \cos\theta \end{pmatrix}, \quad R_y(\theta) = \begin{pmatrix} i\cos\theta & \sin\theta \\ \sin\theta & i\cos\theta \end{pmatrix},$$

$$R_z(\theta) = \begin{pmatrix} e^{i\theta} & 0 \\ 0 & e^{-i\theta} \end{pmatrix},$$

As a generalization we have a rotation around an arbitrary real unit vector $\bar{n} = (n_x, n_y, n_z)$ defined by

$$R_{\bar{n}}(\theta) = e^{-i\theta\bar{n}\cdot\bar{\sigma}/2} = \cos\frac{\theta}{2}(n_x\sigma_x + n_y\sigma_y + n_z\sigma_z).$$

STANDARD (COMPUTATIONAL) BASIS

STANDARD (COMPUTATIONAL) BASIS

A **standard (computational) basis** in a 2^n Hilbert space is the basis $\{|x\rangle\}_{x\in\{0,1\}^n}$.

STANDARD (COMPUTATIONAL) BASIS

A **standard (computational) basis** in a 2^n Hilbert space is the basis $\{|x\rangle\}_{x\in\{0,1\}^n}$.

In particular in 2-dimensional Hilbert space H_2 the **standard (computational basis** contains vectors

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad \text{and} \quad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

STANDARD (COMPUTATIONAL) BASIS

A **standard (computational) basis** in a 2^n Hilbert space is the basis $\{|x\rangle\}_{x\in\{0,1\}^n}$.

In particular in 2-dimensional Hilbert space H_2 the **standard (computational basis** contains vectors

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad \text{and} \quad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

In H_2 of importance is also so called **dual basis** (or $\pm$-basis or Hadamard basis) consisting of states

STANDARD (COMPUTATIONAL) BASIS

A **standard (computational) basis** in a 2^n Hilbert space is the basis $\{|x\rangle\}_{x\in\{0,1\}^n}$.

In particular in 2-dimensional Hilbert space H_2 the **standard (computational basis** contains vectors

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad \text{and} \quad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

In H_2 of importance is also so called **dual basis** (or $\pm$-basis or Hadamard basis) consisting of states

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

STANDARD (COMPUTATIONAL) BASIS

A **standard (computational) basis** in a 2^n Hilbert space is the basis $\{|x\rangle\}_{x\in\{0,1\}^n}$.

In particular in 2-dimensional Hilbert space H_2 the **standard (computational basis** contains vectors

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad \text{and} \quad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

In H_2 of importance is also so called **dual basis** (or $\pm$-basis or Hadamard basis) consisting of states

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

Of importance is that the Hadamard matrix maps the standard basis into dual one and vice verse.

PAULI MATRICES/GATES

Important one-qubit unitary matrices are the following **Pauli matrices**, expressed in the standard basis as follows;

PAULI MATRICES/GATES

Important one-qubit unitary matrices are the following **Pauli matrices**, expressed in the standard basis as follows;

$$\sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix},$$

PAULI MATRICES/GATES

Important one-qubit unitary matrices are the following **Pauli matrices**, expressed in the standard basis as follows;

$$\sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad \sigma_y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix},$$

PAULI MATRICES/GATES

Important one-qubit unitary matrices are the following **Pauli matrices**, expressed in the standard basis as follows;

$$\sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad \sigma_y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad \sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

PAULI MATRICES/GATES

Important one-qubit unitary matrices are the following **Pauli matrices**, expressed in the standard basis as follows;

$$\sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad \sigma_y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad \sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

Observe that Pauli matrices transform a qubit state $|\phi\rangle = \alpha|0\rangle + \beta|1\rangle$ as follows

PAULI MATRICES/GATES

Important one-qubit unitary matrices are the following **Pauli matrices**, expressed in the standard basis as follows;

$$\sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad \sigma_y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad \sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

Observe that Pauli matrices transform a qubit state $|\phi\rangle = \alpha|0\rangle + \beta|1\rangle$ as follows

$$\sigma_x(\alpha|0\rangle + \beta|1\rangle) = \beta|0\rangle + \alpha|1\rangle$$

PAULI MATRICES/GATES

Important one-qubit unitary matrices are the following **Pauli matrices**, expressed in the standard basis as follows;

$$\sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad \sigma_y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad \sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

Observe that Pauli matrices transform a qubit state $|\phi\rangle = \alpha|0\rangle + \beta|1\rangle$ as follows

$$\sigma_x(\alpha|0\rangle + \beta|1\rangle) = \beta|0\rangle + \alpha|1\rangle \quad \sigma_z(\alpha|0\rangle + \beta|1\rangle) = \alpha|0\rangle - \beta|1\rangle$$

PAULI MATRICES/GATES

Important one-qubit unitary matrices are the following **Pauli matrices**, expressed in the standard basis as follows;

$$\sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad \sigma_y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad \sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

Observe that Pauli matrices transform a qubit state $|\phi\rangle = \alpha|0\rangle + \beta|1\rangle$ as follows

$$\sigma_x(\alpha|0\rangle + \beta|1\rangle) = \beta|0\rangle + \alpha|1\rangle \quad \sigma_z(\alpha|0\rangle + \beta|1\rangle) = \alpha|0\rangle - \beta|1\rangle$$

and for $\sigma'_y = \sigma_x\sigma_z$ we have

$$\sigma'_y(|\alpha|0\rangle + \beta|1\rangle) = \beta|0\rangle - \alpha|1\rangle.$$

PAULI MATRICES/GATES

Important one-qubit unitary matrices are the following **Pauli matrices**, expressed in the standard basis as follows;

$$\sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad \sigma_y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad \sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

Observe that Pauli matrices transform a qubit state $|\phi\rangle = \alpha|0\rangle + \beta|1\rangle$ as follows

$$\sigma_x(\alpha|0\rangle + \beta|1\rangle) = \beta|0\rangle + \alpha|1\rangle \quad \sigma_z(\alpha|0\rangle + \beta|1\rangle) = \alpha|0\rangle - \beta|1\rangle$$

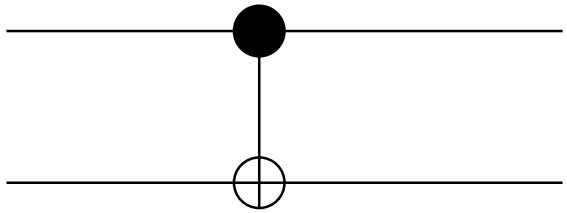
and for $\sigma'_y = \sigma_x\sigma_z$ we have

$$\sigma'_y(|\alpha|0\rangle + \beta|1\rangle) = \beta|0\rangle - \alpha|1\rangle.$$

Operators σ_x, σ_z and σ'_y represent therefore a *bit error*, a *sign error* and a *bit-sign error*.

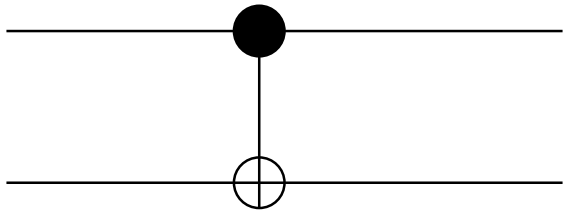
XOR (CNOT) GATE

Unitary matrix for so called XOR-gate (CNOT-gate)



XOR (CNOT) GATE

Unitary matrix for so called XOR-gate (CNOT-gate)

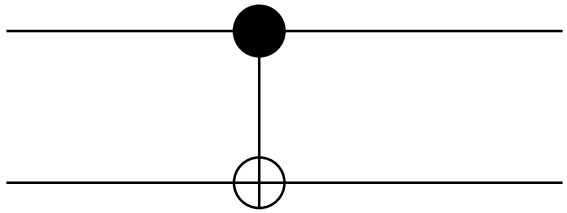


has the form

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}.$$

XOR (CNOT) GATE

Unitary matrix for so called XOR-gate (CNOT-gate)



has the form

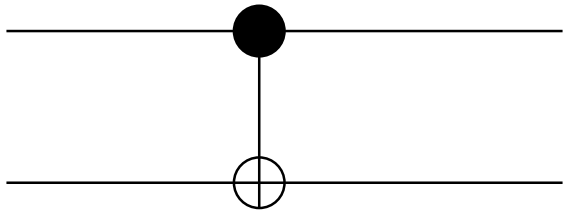
$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}.$$

The CNOT gate maps inputs as follows:

$$|00\rangle \to |00\rangle \quad |01\rangle \to |01\rangle \quad |10\rangle \to |11\rangle \quad |11\rangle \to 10\rangle$$

XOR (CNOT) GATE

Unitary matrix for so called XOR-gate (CNOT-gate)



has the form

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}.$$

The CNOT gate maps inputs as follows:

$$|00\rangle \to |00\rangle \quad |01\rangle \to |01\rangle \quad |10\rangle \to |11\rangle \quad |11\rangle \to 10\rangle$$

That is, if first (control) qubit is $|0\rangle$, then neither first nor second inputs are changed; if first qubit is $|1\rangle$, then the first qubit is not changed, but second is negated.

UNIVERSAL SETS of GATES

UNIVERSAL SETS of GATES

A set U of gates is called universal if each unitary matrix mapping can be realized, with arbitrary precision, by a finite circit consisting of gates from the set U only.

UNIVERSAL SETS of GATES

A set U of gates is called universal if each unitary matrix mapping can be realized, with arbitrary precision, by a finite circit consisting of gates from the set U only. The following set of gates is universal:

UNIVERSAL SETS of GATES

A set U of gates is called universal if each unitary matrix mapping can be realized, with arbitrary precision, by a finite circit consisting of gates from the set U only. The following set of gates is universal:

1. CNOT gate and all one-qubit gates;

UNIVERSAL SETS of GATES

A set U of gates is called universal if each unitary matrix mapping can be realized, with arbitrary precision, by a finite circit consisting of gates from the set U only. The following set of gates is universal:

1. CNOT gate and all one-qubit gates;
2. Three gates:

$$\text{CNOT} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}, H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \sigma_z^{1/4} = \begin{pmatrix} 1 & 0 \\ 0 & e^{\frac{\pi}{4} i} \end{pmatrix}$$

PROBLEMS with CNOT GATE

Out of three gates in the above set of universal gates only the CNOT gate is (very) difficult to implement.

PROBLEMS with CNOT GATE

Out of three gates in the above set of universal gates only the CNOT gate is (very) difficult to implement.

Main reason behind is that the CNOT gate maps some nonentangled states into entangled states exhibiting non-locality.

FROM GATES to UNITARY MATRICES

In general, if a quantum gate G has n inputs and outputs, then for the corresponding unitary matrix of G , with both rows and columns labeled by all n-bit strings, the entry

in the column $x \in \{0,1\}^n$

and

in the row $y \in \{0,1\}^n$

FROM GATES to UNITARY MATRICES

In general, if a quantum gate G has n inputs and outputs, then for the corresponding unitary matrix of G , with both rows and columns labeled by all n-bit strings, the entry

in the column $x \in \{0,1\}^n$

and

in the row $y \in \{0,1\}^n$

is the amplitude for transition, under the mapping G, from the basis state $|x\rangle$ to the basis state $|y\rangle$.

REPRESENTATION of GATES by MATRICES in DIFFERENT BASIS

REPRESENTATION of GATES by MATRICES in DIFFERENT BASIS

Unitary operators have different matrix representations in different bases.

REPRESENTATION of GATES by MATRICES in DIFFERENT BASIS

Unitary operators have different matrix representations in different bases.

For example XOR operator has in the standard basis

$$\{|00\rangle, |01\rangle, |10\rangle, |11\rangle\}$$

representation

REPRESENTATION of GATES by MATRICES in DIFFERENT BASIS

Unitary operators have different matrix representations in different bases.

For example XOR operator has in the standard basis

$$\{|00\rangle, |01\rangle, |10\rangle, |11\rangle\}$$

representation

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

REPRESENTATION of GATES by MATRICES in DIFFERENT BASIS

Unitary operators have different matrix representations in different bases.

For example XOR operator has in the standard basis

$$\{|00\rangle, |01\rangle, |10\rangle, |11\rangle\}$$

representation

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

and in the basis

$$\{|00\rangle, |10\rangle, |01\rangle, |11\rangle\}$$

its representation is

REPRESENTATION of GATES by MATRICES in DIFFERENT BASIS

Unitary operators have different matrix representations in different bases.

For example XOR operator has in the standard basis

$$\{|00\rangle, |01\rangle, |10\rangle, |11\rangle\}$$

representation

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

and in the basis

$$\{|00\rangle, |10\rangle, |01\rangle, |11\rangle\}$$

its representation is

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

CNOT in BEL BASIS

CNOT in BEL BASIS

Representation of CNOT in the Bell basis $\{|\Phi^+\rangle, |\Phi^-\rangle, |\Psi^+\rangle, |\Psi^-\rangle\}$, where

$$|\Phi^+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \qquad |\Phi^-\rangle = \frac{1}{\sqrt{2}}(|00\rangle - |11\rangle)$$

$$|\Psi^+\rangle = \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle) \qquad |\Psi^-\rangle = \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle)$$

CNOT in BEL BASIS

Representation of CNOT in the Bell basis $\{|\Phi^+\rangle, |\Phi^-\rangle, |\Psi^+\rangle, |\Psi^-\rangle\}$, where

$$|\Phi^+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \qquad |\Phi^-\rangle = \frac{1}{\sqrt{2}}(|00\rangle - |11\rangle)$$

$$|\Psi^+\rangle = \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle) \qquad |\Psi^-\rangle = \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle)$$

has the form

CNOT in BEL BASIS

Representation of CNOT in the Bell basis $\{|\Phi^+\rangle, |\Phi^-\rangle, |\Psi^+\rangle, |\Psi^-\rangle\}$, where

$$|\Phi^+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \quad |\Phi^-\rangle = \frac{1}{\sqrt{2}}(|00\rangle - |11\rangle)$$

$$|\Psi^+\rangle = \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle) \quad |\Psi^-\rangle = \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle)$$

has the form

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

SCHRÖDINGER EQUATION

Evolution in a quantum system is described by **Schrödinger linear equation**

$$i\hbar\frac{\partial\psi(t)}{\partial t} = H(t)\psi(t),$$

where $H(t)$ is a Hamiltonian (Hermitian operator representing total energy of the system), from which it follows that

$$\psi(t) = e^{-\frac{i}{\hbar}H(t)\psi(0)}$$

, where $e^{-\frac{i}{\hbar}H(t)}$ is a unitary matrix, and therefore that at a discretized evolution a (computation) step of a quantum system is performed by a multiplication, of the state vector, by a **unitary operator**.

SCHRÖDINGER EQUATION

Evolution in a quantum system is described by **Schrödinger linear equation**

$$i\hbar\frac{\partial\psi(t)}{\partial t} = H(t)\psi(t),$$

where $H(t)$ is a Hamiltonian (Hermitian operator representing total energy of the system), from which it follows that

$$\psi(t) = e^{-\frac{i}{\hbar}H(t)\psi(0)}$$

, where $e^{-\frac{i}{\hbar}H(t)}$ is a unitary matrix, and therefore that at a discretized evolution a (computation) step of a quantum system is performed by a multiplication, of the state vector, by a **unitary operator**.

In other words, if Hamiltonian is constant, then a step of evolution of a state $|\psi\rangle$ is a multiplication by a **unitary matrix** A of a vector $|\psi\rangle$, i.e.

$$A|\psi\rangle$$

HAMILTONIAN for the CNOT GATE

For the Hamiltonian

$$H = \frac{\pi\hbar}{2}\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & -1 & 1 \end{pmatrix} = \frac{\pi\hbar}{2} V$$

HAMILTONIAN for the CNOT GATE

For the Hamiltonian

$$H = \frac{\pi\hbar}{2}\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & -1 & 1 \end{pmatrix} = \frac{\pi\hbar}{2}V$$

the Schrödinger equation

$$i\hbar\frac{\partial U(t)}{\partial t} = HU(t)$$

has the solution

$$U(t) = e^{-\frac{i}{\hbar}Ht} =$$

HAMILTONIAN for the CNOT GATE

For the Hamiltonian

$$H = \frac{\pi\hbar}{2}\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & -1 & 1 \end{pmatrix} = \frac{\pi\hbar}{2} V$$

the Schrödinger equation

$$i\hbar\frac{\partial U(t)}{\partial t} = HU(t)$$

has the solution

$$U(t) = e^{-\frac{i}{\hbar}Ht} = \sum_{k=0}^{\infty} \frac{(-\frac{i\pi}{2})^k V^k t^k}{k!} =$$

HAMILTONIAN for the CNOT GATE

For the Hamiltonian

$$H = \frac{\pi\hbar}{2}\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & -1 & 1 \end{pmatrix} = \frac{\pi\hbar}{2} V$$

the Schrödinger equation

$$i\hbar\frac{\partial U(t)}{\partial t} = HU(t)$$

has the solution

$$U(t) = e^{-\frac{i}{\hbar}Ht} = \sum_{k=0}^{\infty}\frac{(-\frac{i\pi}{2})^k V^k t^k}{k!} = I + \frac{1}{2}\sum_{k=1}^{\infty}\frac{(-\pi i t)^k}{k!} V$$

because $V^k = 2^{k-1}V$

HAMILTONIAN for the CNOT GATE

For the Hamiltonian

$$H = \frac{\pi\hbar}{2}\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & -1 & 1 \end{pmatrix} = \frac{\pi\hbar}{2} V$$

the Schrödinger equation

$$i\hbar\frac{\partial U(t)}{\partial t} = HU(t)$$

has the solution

$$U(t) = e^{-\frac{i}{\hbar}Ht} = \sum_{k=0}^{\infty}\frac{(-\frac{i\pi}{2})^k V^k t^k}{k!} = I + \frac{1}{2}\sum_{k=1}^{\infty}\frac{(-\pi i t)^k}{k!} V$$

because $V^k = 2^{k-1}V$ and therefore for $t = 1$,

HAMILTONIAN for the CNOT GATE

For the Hamiltonian

$$H = \frac{\pi\hbar}{2}\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & -1 & 1 \end{pmatrix} = \frac{\pi\hbar}{2} V$$

the Schrödinger equation

$$i\hbar\frac{\partial U(t)}{\partial t} = HU(t)$$

has the solution

$$U(t) = e^{-\frac{i}{\hbar}Ht} = \sum_{k=0}^{\infty}\frac{(-\frac{i\pi}{2})^k V^k t^k}{k!} = I + \frac{1}{2}\sum_{k=1}^{\infty}\frac{(-\pi i t)^k}{k!} V$$

because $V^k = 2^{k-1}V$ and therefore for $t = 1$,

$$e^{-\frac{i\pi}{2}V} =$$

HAMILTONIAN for the CNOT GATE

For the Hamiltonian

$$H = \frac{\pi\hbar}{2}\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & -1 & 1 \end{pmatrix} = \frac{\pi\hbar}{2} V$$

the Schrödinger equation

$$i\hbar\frac{\partial U(t)}{\partial t} = HU(t)$$

has the solution

$$U(t) = e^{-\frac{i}{\hbar}Ht} = \sum_{k=0}^{\infty}\frac{(-\frac{i\pi}{2})^k V^k t^k}{k!} = I + \frac{1}{2}\sum_{k=1}^{\infty}\frac{(-\pi i t)^k}{k!} V$$

because $V^k = 2^{k-1}V$ and therefore for $t = 1$,

$$e^{-\frac{i\pi}{2}V} = I + \frac{1}{2}(e^{-i\pi} - 1)V =$$

HAMILTONIAN for the CNOT GATE

For the Hamiltonian

$$H = \frac{\pi\hbar}{2}\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & -1 & 1 \end{pmatrix} = \frac{\pi\hbar}{2} V$$

the Schrödinger equation

$$i\hbar\frac{\partial U(t)}{\partial t} = HU(t)$$

has the solution

$$U(t) = e^{-\frac{i}{\hbar}Ht} = \sum_{k=0}^{\infty}\frac{(-\frac{i\pi}{2})^k V^k t^k}{k!} = I + \frac{1}{2}\sum_{k=1}^{\infty}\frac{(-\pi i t)^k}{k!} V$$

because $V^k = 2^{k-1}V$ and therefore for $t = 1$,

$$e^{-\frac{i\pi}{2}V} = I + \frac{1}{2}(e^{-i\pi} - 1)V = I - V = CNOT.$$

QUANTUM CIRCUITS

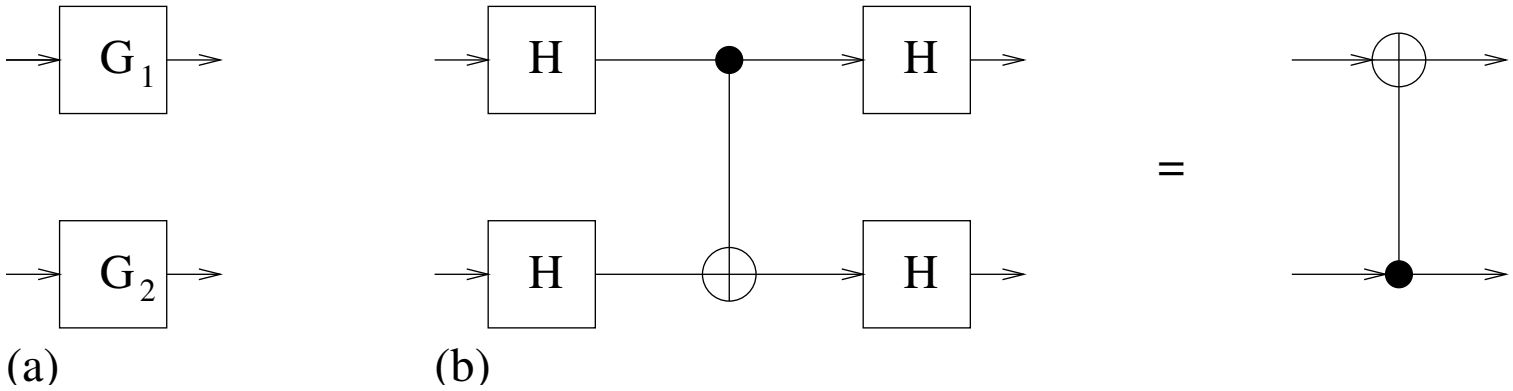
INVERSE CNOT-GATE CIRCUIT

INVERSE CNOT-GATE CIRCUIT

An implementation of the inverse of the CNOT gate.

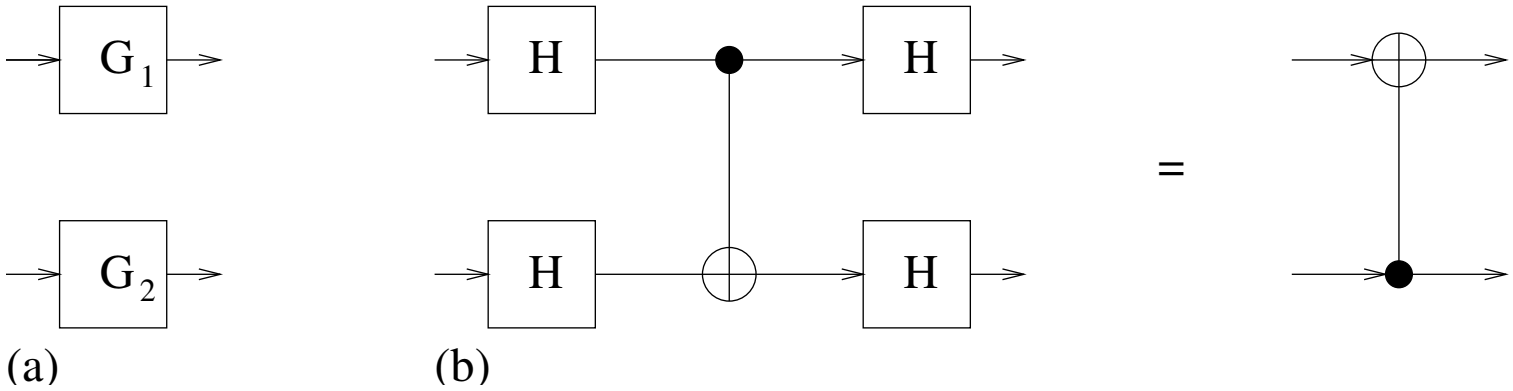
INVERSE CNOT-GATE CIRCUIT

An implementation of the inverse of the CNOT gate.



INVERSE CNOT-GATE CIRCUIT

An implementation of the inverse of the CNOT gate.



The processing in the network on the left side of the identity in Figure 33b for the input $|0\rangle|1\rangle$ can be depicted as shown on next slide:

$$|0\rangle|1\rangle \quad \overset{H-\text{gates}}{\longrightarrow} \quad |0'\rangle|1'\rangle$$

$$|0\rangle|1\rangle \quad \stackrel{H-\text{gates}}{\longrightarrow} \quad |0'\rangle|1'\rangle$$

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

$$|0\rangle|1\rangle \quad \stackrel{H-\text{gates}}{\longrightarrow} \quad |0'\rangle|1'\rangle$$

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

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

$$|0\rangle|1\rangle \quad \overset{H-\text{gates}}{\longrightarrow} \quad |0'\rangle|1'\rangle$$

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

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

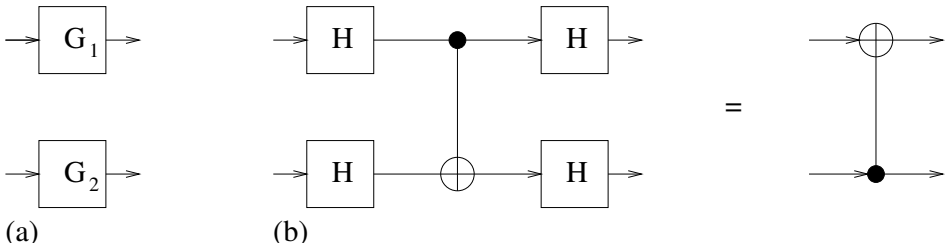
$$\overset{\text{CNOT gate}}{\longrightarrow}$$

$$
\begin{aligned}
|0\rangle|1\rangle \quad & \stackrel{H-\text{gates}}{\longrightarrow} \quad |0'\rangle|1'\rangle \\
& = \quad \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \\
& = \quad \frac{1}{2}(|0\rangle|0\rangle + |1\rangle|0\rangle - |0\rangle|1\rangle - |1\rangle|1\rangle) \\
& \stackrel{\text{CNOT gate}}{\longrightarrow} \quad \frac{1}{2}(|0\rangle|0\rangle + |1\rangle|1\rangle - |0\rangle|1\rangle - |1\rangle|0\rangle)
\end{aligned}
$$

$$
\begin{aligned}
|0\rangle|1\rangle \quad & \stackrel{H-\text{gates}}{\longrightarrow} \quad |0'\rangle|1'\rangle \\
& = \quad \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \\
& = \quad \frac{1}{2}(|0\rangle|0\rangle + |1\rangle|0\rangle - |0\rangle|1\rangle - |1\rangle|1\rangle) \\
& \stackrel{\text{CNOT gate}}{\longrightarrow} \quad \frac{1}{2}(|0\rangle|0\rangle + |1\rangle|1\rangle - |0\rangle|1\rangle - |1\rangle|0\rangle) \\
& = \quad \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) =
\end{aligned}
$$

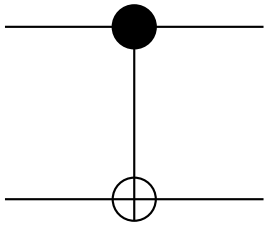
$$
\begin{aligned}
|0\rangle|1\rangle \quad & \stackrel{H-\text{gates}}{\longrightarrow} \quad |0'\rangle|1'\rangle \\
& = \quad \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \\
& = \quad \frac{1}{2}(|0\rangle|0\rangle + |1\rangle|0\rangle - |0\rangle|1\rangle - |1\rangle|1\rangle) \\
& \stackrel{\text{CNOT gate}}{\longrightarrow} \quad \frac{1}{2}(|0\rangle|0\rangle + |1\rangle|1\rangle - |0\rangle|1\rangle - |1\rangle|0\rangle) \\
& = \quad \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) = |1'\rangle|1'\rangle
\end{aligned}
$$

$$
\begin{aligned}
|0\rangle|1\rangle \quad & \xrightarrow{H-\text{gates}} \quad |0'\rangle|1'\rangle \\
& = \quad \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \\
& = \quad \frac{1}{2}(|0\rangle|0\rangle + |1\rangle|0\rangle - |0\rangle|1\rangle - |1\rangle|1\rangle) \\
& \xrightarrow{\text{CNOT gate}} \quad \frac{1}{2}(|0\rangle|0\rangle + |1\rangle|1\rangle - |0\rangle|1\rangle - |1\rangle|0\rangle) \\
& = \quad \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) = |1'\rangle|1'\rangle \\
& \xrightarrow{H \text{ gates}} \quad |1\rangle|1\rangle.
\end{aligned}
$$

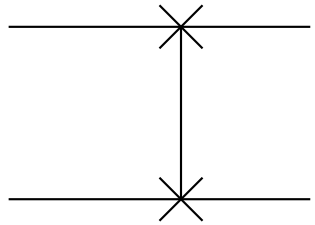


GRAPHICAL REPRESENTATION of BASIC GATES

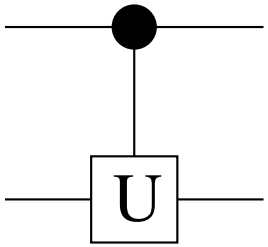
CNOT–gate



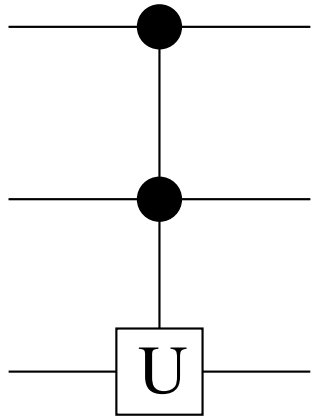
SWAP–gate



Λ(U)–gate

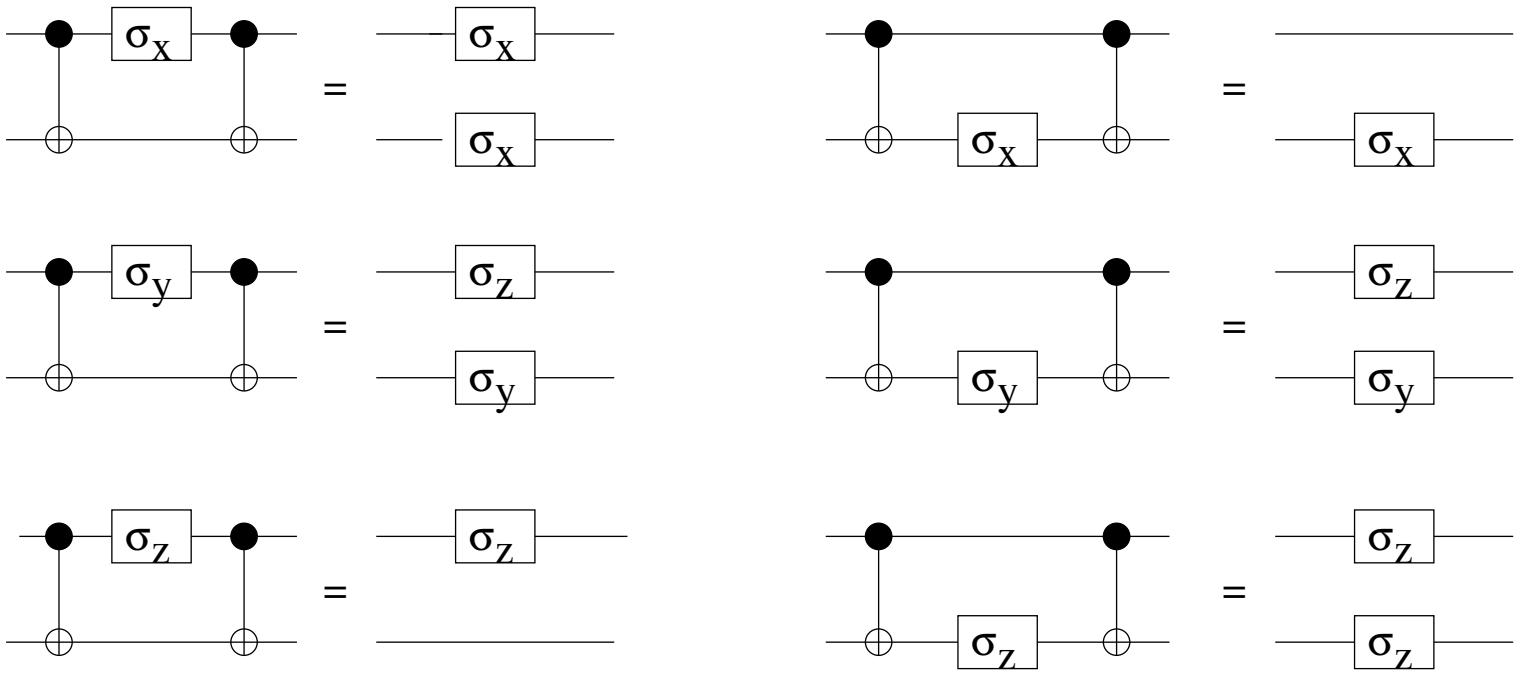


Λ_2 (U)–gate



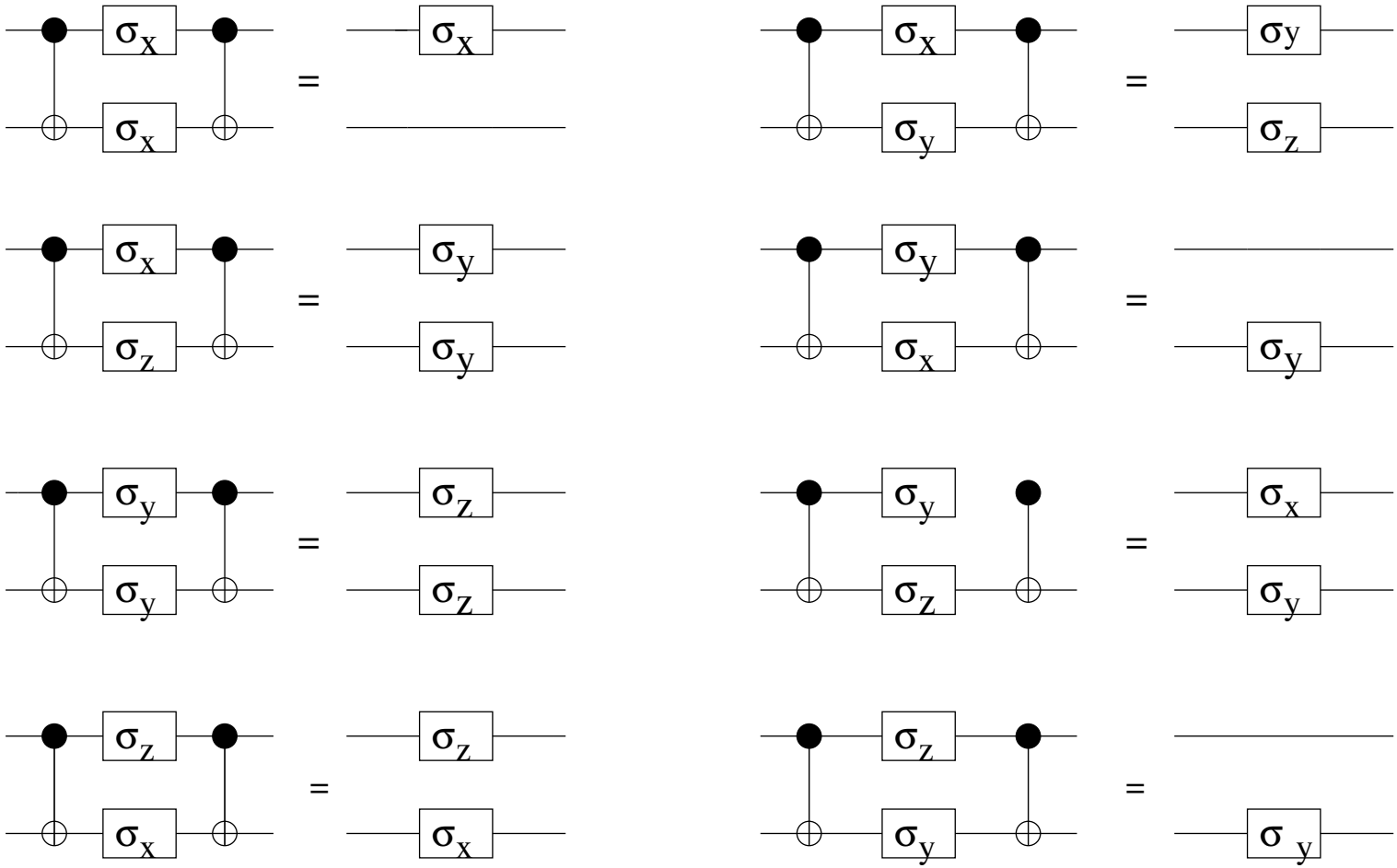
SOME USEFUL IDENTITIES - I.

Several simple identities between elementary gates are surprisingly useful.



SOME USEFUL IDENTITIES - II.

Several simple identities between elementary gates are surprisingly useful.

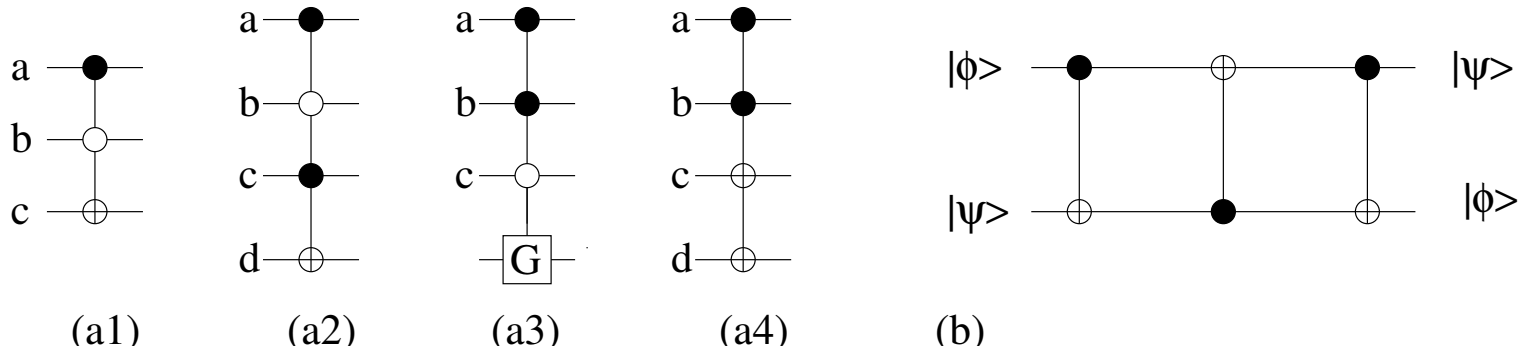


GENERALIZED CNOT GATES

In the following figure several generalizations of the CNOT gate are shown as well as a circuit to flip qubits.

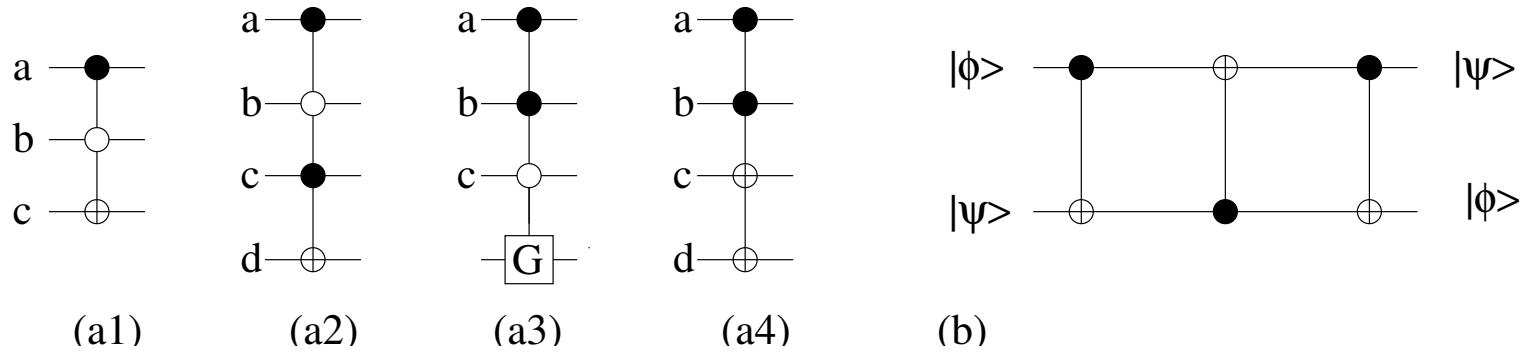
GENERALIZED CNOT GATES

In the following figure several generalizations of the CNOT gate are shown as well as a circuit to flip qubits.



GENERALIZED CNOT GATES

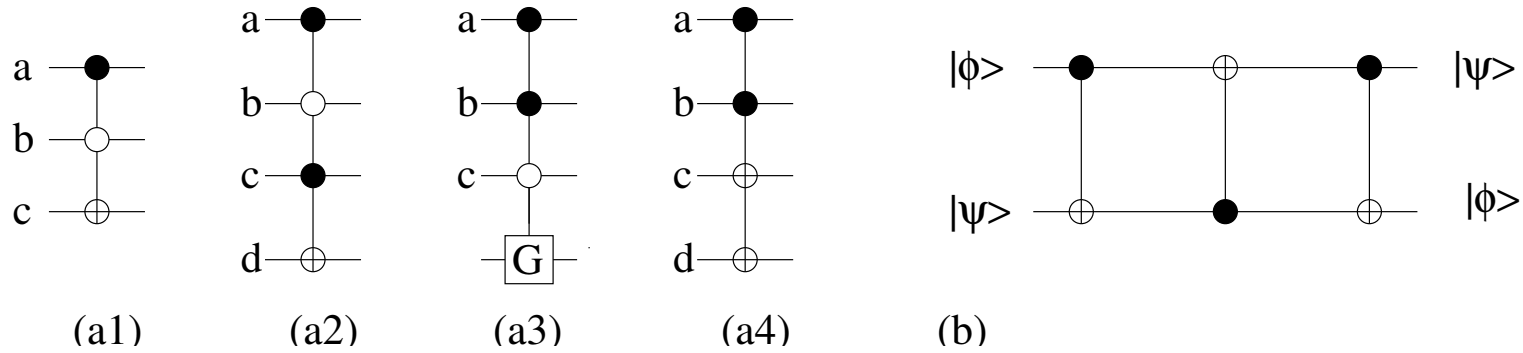
In the following figure several generalizations of the CNOT gate are shown as well as a circuit to flip qubits.



The circuit in Figure b realizes flipping of qubits.

GENERALIZED CNOT GATES

In the following figure several generalizations of the CNOT gate are shown as well as a circuit to flip qubits.

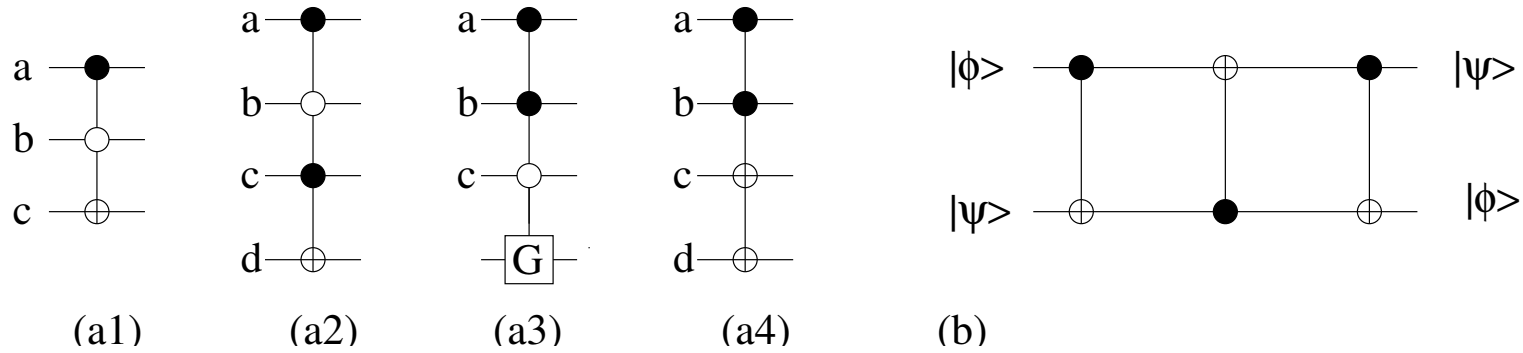


The circuit in Figure b realizes flipping of qubits.

To see that, denote I_{jk} the matrix obtained from the unit matrix of degree 4 by exchanging j-th and h-th columns (i.e. XOR=I_{34}).

GENERALIZED CNOT GATES

In the following figure several generalizations of the CNOT gate are shown as well as a circuit to flip qubits.



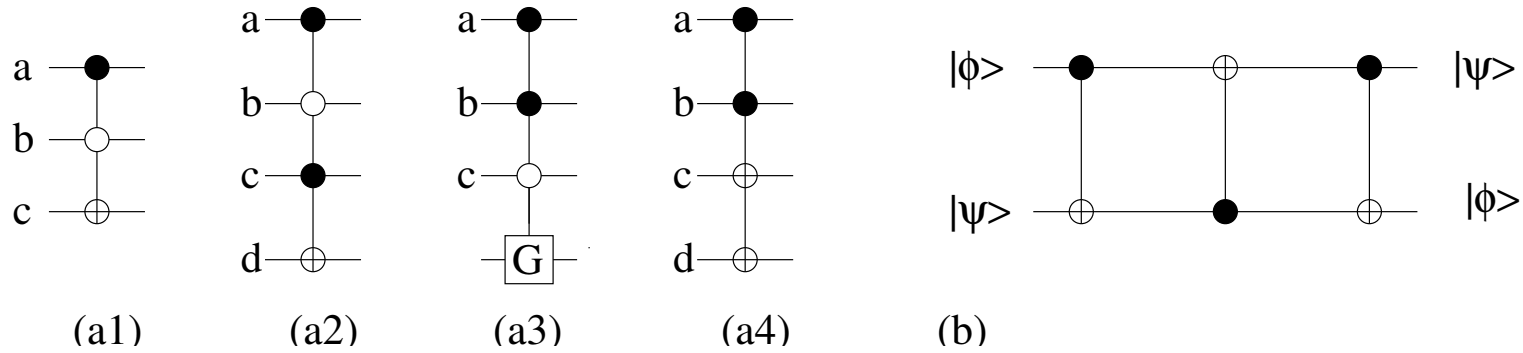
The circuit in Figure b realizes flipping of qubits.

To see that, denote I_{jk} the matrix obtained from the unit matrix of degree 4 by exchanging j-th and h-th columns (i.e. XOR=I_{34}).

If $|\phi\rangle = \alpha|0\rangle + \beta|1\rangle$, $|\psi\rangle = \alpha'|0\rangle + \beta'|1\rangle$, then computation by the circuit in Figure b, gate by gate, corresponds to the following matrix computation:

GENERALIZED CNOT GATES

In the following figure several generalizations of the CNOT gate are shown as well as a circuit to flip qubits.



The circuit in Figure b realizes flipping of qubits.

To see that, denote I_{jk} the matrix obtained from the unit matrix of degree 4 by exchanging j-th and h-th columns (i.e. XOR=I_{34}).

If $|\phi\rangle = \alpha|0\rangle + \beta|1\rangle$, $|\psi\rangle = \alpha'|0\rangle + \beta'|1\rangle$, then computation by the circuit in Figure b, gate by gate, corresponds to the following matrix computation:

$$I_{34} I_{24} I_{34} \begin{pmatrix} \alpha\alpha' \\ \alpha\beta' \\ \beta\alpha' \\ \beta\beta' \end{pmatrix} = I_{34} I_{24} \begin{pmatrix} \alpha\alpha' \\ \alpha\beta' \\ \beta\beta' \\ \beta\alpha' \end{pmatrix} = I_{34} \begin{pmatrix} \alpha\alpha' \\ \beta\alpha' \\ \beta\beta' \\ \alpha\beta' \end{pmatrix} = \begin{pmatrix} \alpha\alpha' \\ \beta\alpha' \\ \alpha\beta' \\ \beta\beta' \end{pmatrix} = \begin{pmatrix} \alpha'\alpha \\ \alpha'\beta \\ \beta'\alpha \\ \beta'\beta \end{pmatrix}$$

PERMUTATION CIRCUIT

PERMUTATION CIRCUIT

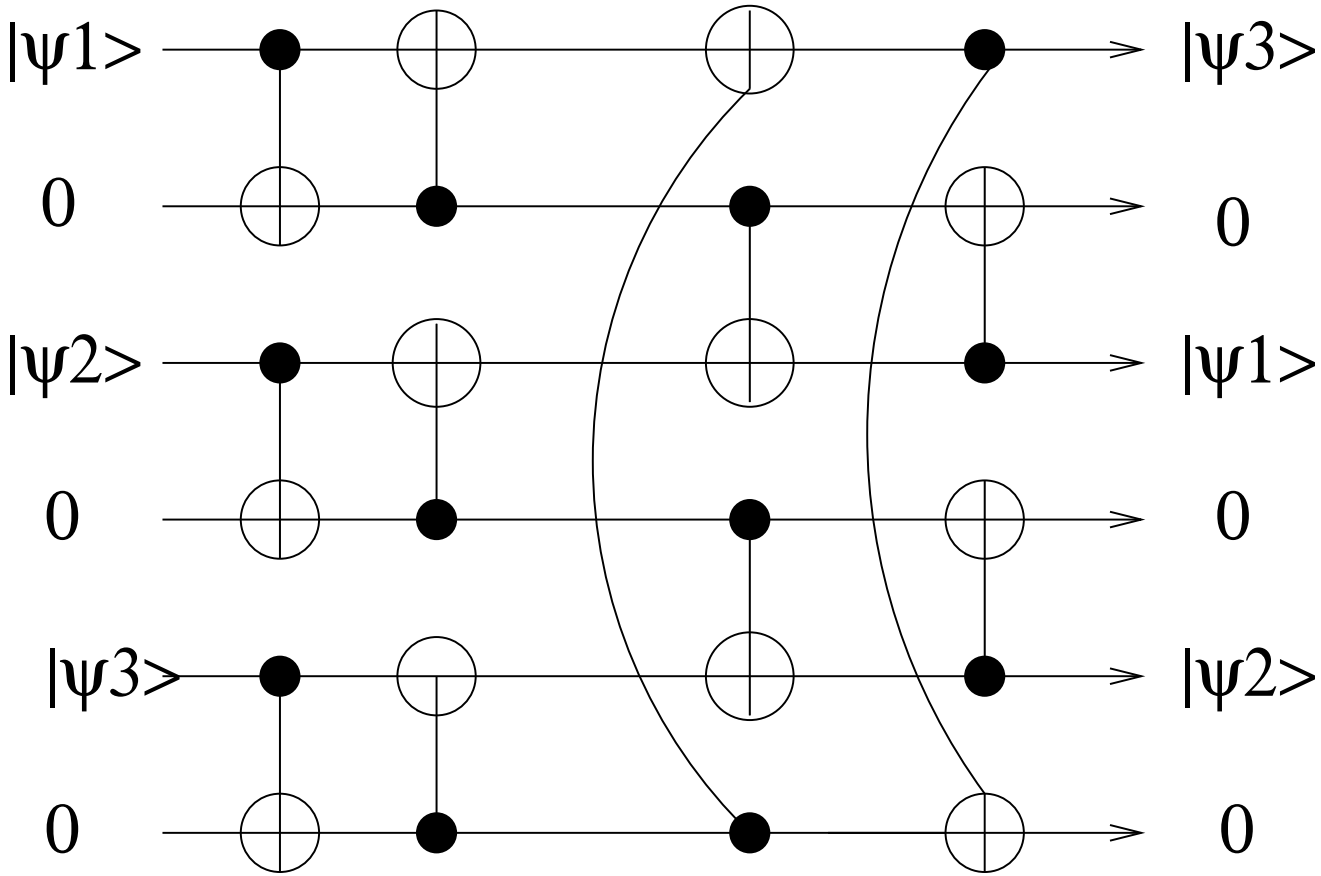
Using several copies of the circuit to flip (or to transpose) two qubits one can realize any permutation of qubits.

PERMUTATION CIRCUIT

Using several copies of the circuit to flip (or to transpose) two qubits one can realize any permutation of qubits. Using such a method one needs 6 gate-steps to perform permutation shown in the following figure, where such a permutation is realized, using a more complex circuit, with three ancilla qubits, but in only four gate-steps.

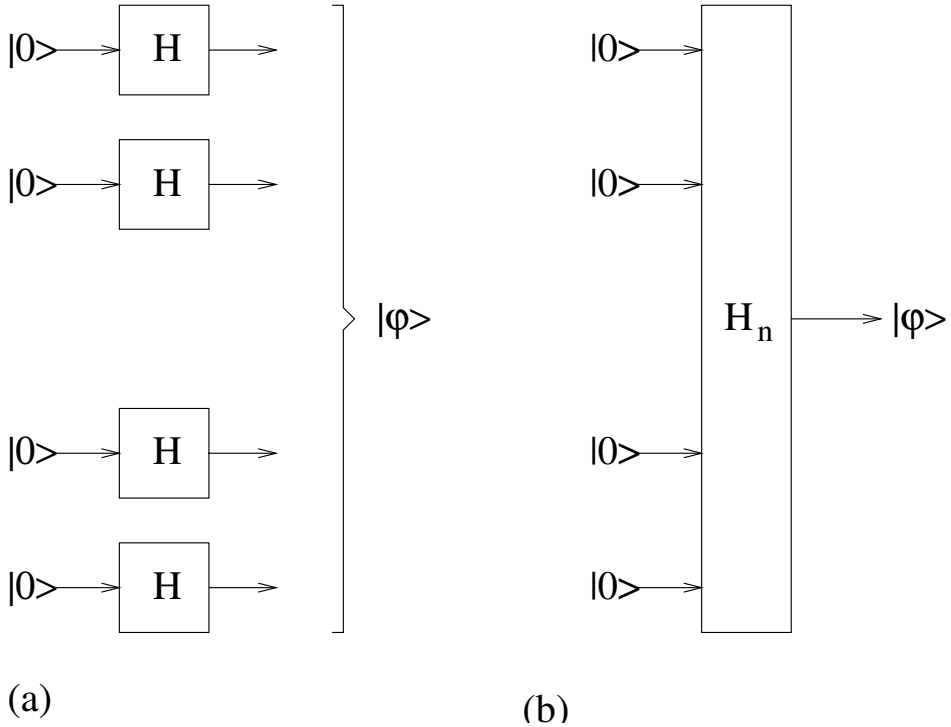
PERMUTATION CIRCUIT

Using several copies of the circuit to flip (or to transpose) two qubits one can realize any permutation of qubits. Using such a method one needs 6 gate-steps to perform permutation shown in the following figure, where such a permutation is realized, using a more complex circuit, with three ancilla qubits, but in only four gate-steps.



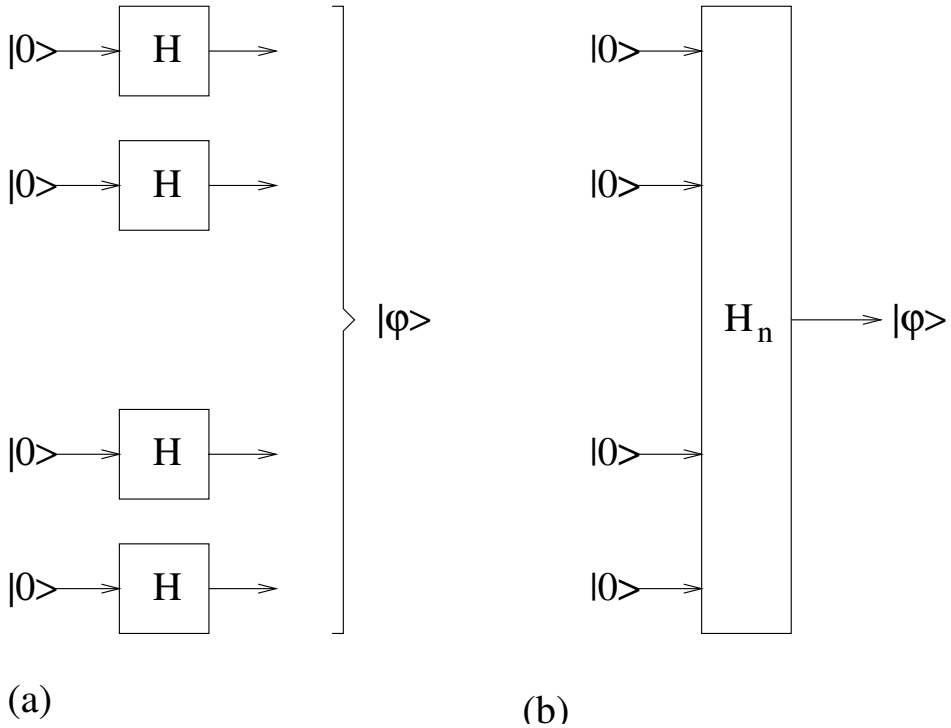
HADAMARD GATE

The Hadamard transform H_n is implemented by the circuit in Figure a, and Figure b contains the usual notation for the circuit for H_n.



HADAMARD GATE

The Hadamard transform H_n is implemented by the circuit in Figure a, and Figure b contains the usual notation for the circuit for H_n.



The Hadamard circuit/gate H_n when applied to the state $|0^{(n)}\rangle$ provides as the outcome the state

$$|\phi\rangle = \frac{1}{\sqrt{2^n}} \sum_{i=0}^{2^n-1} |i\rangle.$$

HADAMARD GATE APPLIED TO BASIS STATES

For an $x \in \{0,1\}^n$ it holds

$$H_n(|x\rangle) = \frac{1}{\sqrt{2^n}} \sum_{y \in \{0,1\}^n} (-1)^{x \cdot y} |y\rangle$$

where $x \cdot y$ stands for the inner product of strings x and y, that is their bit-wise xor-multiplications and then the addition modulo 2.

SIMPLE QUANTUM ALGORITHMS

QUANTUM PARALLELISM

If

$$f : \{0, 1, \ldots, 2^n - 1\} \Longrightarrow \{0, 1, \ldots, 2^n - 1\}$$

QUANTUM PARALLELISM

If

$$f : \{0, 1, \ldots, 2^n - 1\} \Longrightarrow \{0, 1, \ldots, 2^n - 1\}$$

then the mapping

$$f' : (x, b) \Longrightarrow (x, b \oplus f(x)),$$

where $x, b \in \{0, 1, \ldots, 2^n - 1\}$

QUANTUM PARALLELISM

If

$$f : \{0, 1, \ldots, 2^n - 1\} \Longrightarrow \{0, 1, \ldots, 2^n - 1\}$$

then the mapping

$$f' : (x, b) \Longrightarrow (x, b \oplus f(x)),$$

where $x, b \in \{0, 1, \ldots, 2^n - 1\}$ is one-to-one

QUANTUM PARALLELISM

If

$$f : \{0, 1, \ldots, 2^n - 1\} \Longrightarrow \{0, 1, \ldots, 2^n - 1\}$$

then the mapping

$$f' : (x, b) \Longrightarrow (x, b \oplus f(x)),$$

where $x, b \in \{0, 1, \ldots, 2^n - 1\}$ is one-to-one and therefore there is a unitary transformation U_f such that.

$$U_f(|x\rangle|0\rangle) \Longrightarrow |x\rangle|f(x)\rangle$$

QUANTUM PARALLELISM

If

$$f : \{0, 1, \ldots, 2^n - 1\} \Longrightarrow \{0, 1, \ldots, 2^n - 1\}$$

then the mapping

$$f' : (x, b) \Longrightarrow (x, b \oplus f(x)),$$

where $x, b \in \{0, 1, \ldots, 2^n - 1\}$ is one-to-one and therefore there is a unitary transformation U_f such that.

$$U_f(|x\rangle|0\rangle) \Longrightarrow |x\rangle|f(x)\rangle$$

Let

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

QUANTUM PARALLELISM

If

$$f : \{0, 1, \ldots, 2^n - 1\} \Longrightarrow \{0, 1, \ldots, 2^n - 1\}$$

then the mapping

$$f' : (x, b) \Longrightarrow (x, b \oplus f(x)),$$

where $x, b \in \{0, 1, \ldots, 2^n - 1\}$ is one-to-one and therefore there is a unitary transformation U_f such that.

$$U_f(|x\rangle|0\rangle) \Longrightarrow |x\rangle|f(x)\rangle$$

Let

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

With a **single application** of the mapping U_f we get

$$U_f|\psi\rangle = \frac{1}{\sqrt{2^n}} \sum_{i=0}^{2^n-1} |i\rangle|f(i)\rangle$$

Hence, IN A SINGLE COMPUTATIONAL STEP 2^n VALUES OF f ARE "COMPUTED"! - in some sense.

IMPACTS of PROJECTIVE MEASUREMENTS

If we "measure" second register of the state

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

in the standard basis,

IMPACTS of PROJECTIVE MEASUREMENTS

If we "measure" second register of the state

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

in the standard basis, then $|\phi\rangle$ collapses into one of the states

$$|\phi_y\rangle = \frac{1}{\sqrt{k_y}} \sum_{\{x \mid f(x)=y\}} |x\rangle|y\rangle,$$

IMPACTS of PROJECTIVE MEASUREMENTS

If we "measure" second register of the state

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

in the standard basis, then $|\phi\rangle$ collapses into one of the states

$$|\phi_y\rangle = \frac{1}{\sqrt{k_y}} \sum_{\{x \mid f(x)=y\}} |x\rangle|y\rangle,$$

where

- y is in the range of the values of the function f.

IMPACTS of PROJECTIVE MEASUREMENTS

If we "measure" second register of the state

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

in the standard basis, then $|\phi\rangle$ collapses into one of the states

$$|\phi_y\rangle = \frac{1}{\sqrt{k_y}} \sum_{\{x \,|\, f(x)=y\}} |x\rangle|y\rangle,$$

where

- y is in the range of the values of the function f.
- $k_y = |\{x \,|\, f(x) = y\}|$.

IMPACTS of PROJECTIVE MEASUREMENTS

If we "measure" second register of the state

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

in the standard basis, then $|\phi\rangle$ collapses into one of the states

$$|\phi_y\rangle = \frac{1}{\sqrt{k_y}} \sum_{\{x \,|\, f(x)=y\}} |x\rangle|y\rangle,$$

where

- y is in the range of the values of the function f.
- $k_y = |\{x \,|\, f(x) = y\}|$.

The collapse into the state $|\phi_y\rangle$ happens with the probability

$$\frac{k_y}{2^n}$$

IMPACTS of PROJECTIVE MEASUREMENTS

If we "measure" second register of the state

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

in the standard basis, then $|\phi\rangle$ collapses into one of the states

$$|\phi_y\rangle = \frac{1}{\sqrt{k_y}} \sum_{\{x \,|\, f(x)=y\}} |x\rangle|y\rangle,$$

where

- y is in the range of the values of the function f.
- $k_y = |\{x \,|\, f(x) = y\}|$.

The collapse into the state $|\phi_y\rangle$ happens with the probability

$$\frac{k_y}{2^n}$$

and into the classical world one gets information which of y in the range of f, in the second register, has been (randomly) chosen.

IMPACTS of PROJECTIVE MEASUREMENTS

If we "measure" second register of the state

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

in the standard basis, then $|\phi\rangle$ collapses into one of the states

$$|\phi_y\rangle = \frac{1}{\sqrt{k_y}} \sum_{\{x \,|\, f(x)=y\}} |x\rangle|y\rangle,$$

where

- y is in the range of the values of the function f.
- $k_y = |\{x \,|\, f(x) = y\}|$.

The collapse into the state $|\phi_y\rangle$ happens with the probability

$$\frac{k_y}{2^n}$$

and into the classical world one gets information which of y in the range of f, in the second register, has been (randomly) chosen.

This fact we usually interpret that y is the (classical) result of the measurement of the second register of the state $|\phi\rangle$ with respect to the standard basis.

U_f VERSUS V_f OPERATORS

U_f VERSUS V_f OPERATORS

Another useful operator related to functions

$$f : \{0, 1, \ldots, 2^n - 1\} \rightarrow \{0, 1\}$$

is the operator

$$V_f|x\rangle \rightarrow (-1)^{f(x)}|x\rangle,$$

where $x \in \{0, 1, 2, \ldots, 2^n - 1\}$,

U_f VERSUS V_f OPERATORS

Another useful operator related to functions

$$f : \{0, 1, \ldots, 2^n - 1\} \rightarrow \{0, 1\}$$

is the operator

$$V_f|x\rangle \rightarrow (-1)^{f(x)}|x\rangle,$$

where $x \in \{0, 1, 2, \ldots, 2^n - 1\}$, which can be expressed using the operator

$$U_f : |x, b\rangle \rightarrow |x, b \oplus f(x)\rangle$$

and one additional qubit,

U_f VERSUS V_f OPERATORS

Another useful operator related to functions

$$f : \{0, 1, \ldots, 2^n - 1\} \rightarrow \{0, 1\}$$

is the operator

$$V_f|x\rangle \rightarrow (-1)^{f(x)}|x\rangle,$$

where $x \in \{0, 1, 2, \ldots, 2^n - 1\}$, which can be expressed using the operator

$$U_f : |x, b\rangle \rightarrow |x, b \oplus f(x)\rangle$$

and one additional qubit, called again **ancilla**, in the state $\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$ as follows

U_f VERSUS V_f OPERATORS

Another useful operator related to functions

$$f : \{0, 1, \ldots, 2^n - 1\} \rightarrow \{0, 1\}$$

is the operator

$$V_f|x\rangle \rightarrow (-1)^{f(x)}|x\rangle,$$

where $x \in \{0, 1, 2, \ldots, 2^n - 1\}$, which can be expressed using the operator

$$U_f : |x, b\rangle \rightarrow |x, b \oplus f(x)\rangle$$

and one additional qubit, called again **ancilla**, in the state $\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$ as follows

$$\begin{aligned} U_f|x, \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)\rangle &= \frac{1}{\sqrt{2}}(|x, 0 \oplus f(x)\rangle - |x, 1 \oplus f(x)\rangle \\ &= (-1)^{f(x)}|x\rangle \otimes \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \end{aligned}$$

DEUTSCH PROBLEM-RANDOMIZED SOLUTION

Given a function $f : \{0,1\} \to \{0,1\}$, as a **black box**, the task is to determine whether f is constant or balanced.

DEUTSCH PROBLEM-RANDOMIZED SOLUTION

Given a function $f : \{0,1\} \to \{0,1\}$, as a **black box**, the task is to determine whether f is constant or balanced.

To solve the problem:

In classical computing 2 calls of f are needed.

DEUTSCH PROBLEM-RANDOMIZED SOLUTION

Given a function $f : \{0,1\} \to \{0,1\}$, as a **black box**, the task is to determine whether f is constant or balanced.

To solve the problem:

In classical computing 2 calls of f are needed.

In quantum computing 1 call of f is sufficient.

DEUTSCH PROBLEM-RANDOMIZED SOLUTION

Given a function $f : \{0,1\} \to \{0,1\}$, as a **black box**, the task is to determine whether f is constant or balanced.

To solve the problem:

In classical computing 2 calls of f are needed.

In quantum computing 1 call of f is sufficient.

Quantum algorithm presented below solves the problem with probability $\frac{1}{2}$ in such a way that we know whether the answer is correct. Since

$$U_f : (\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)|0\rangle) \to \frac{1}{\sqrt{2}}(|0, f(0)\rangle + |1, f(1)\rangle),$$

DEUTSCH PROBLEM-RANDOMIZED SOLUTION

Given a function $f : \{0,1\} \to \{0,1\}$, as a **black box**, the task is to determine whether f is constant or balanced.

To solve the problem:

In classical computing 2 calls of f are needed.

In quantum computing 1 call of f is sufficient.

Quantum algorithm presented below solves the problem with probability $\frac{1}{2}$ in such a way that we know whether the answer is correct. Since

$$U_f : (\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)|0\rangle) \to \frac{1}{\sqrt{2}}(|0, f(0)\rangle + |1, f(1)\rangle),$$

the result can be written, in the standard and dual basis, as follows:

DEUTSCH PROBLEM-RANDOMIZED SOLUTION

Given a function $f : \{0,1\} \to \{0,1\}$, as a **black box**, the task is to determine whether f is constant or balanced.

To solve the problem:

In classical computing 2 calls of f are needed.

In quantum computing 1 call of f is sufficient.

Quantum algorithm presented below solves the problem with probability $\frac{1}{2}$ in such a way that we know whether the answer is correct. Since

$$U_f : (\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)|0\rangle) \to \frac{1}{\sqrt{2}}(|0, f(0)\rangle + |1, f(1)\rangle),$$

the result can be written, in the standard and dual basis, as follows:

if f is constant:

DEUTSCH PROBLEM-RANDOMIZED SOLUTION

Given a function $f : \{0,1\} \to \{0,1\}$, as a **black box**, the task is to determine whether f is constant or balanced.

To solve the problem:

In classical computing 2 calls of f are needed.

In quantum computing 1 call of f is sufficient.

Quantum algorithm presented below solves the problem with probability $\frac{1}{2}$ in such a way that we know whether the answer is correct. Since

$$U_f : (\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)|0\rangle) \to \frac{1}{\sqrt{2}}(|0, f(0)\rangle + |1, f(1)\rangle),$$

the result can be written, in the standard and dual basis, as follows:

if f is constant:

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

DEUTSCH PROBLEM-RANDOMIZED SOLUTION

Given a function $f : \{0,1\} \to \{0,1\}$, as a **black box**, the task is to determine whether f is constant or balanced.
To solve the problem:

In classical computing 2 calls of f are needed.

In quantum computing 1 call of f is sufficient.

Quantum algorithm presented below solves the problem with probability $\frac{1}{2}$ in such a way that we know whether the answer is correct. Since

$$U_f : (\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)|0\rangle) \to \frac{1}{\sqrt{2}}(|0, f(0)\rangle + |1, f(1)\rangle),$$

the result can be written, in the standard and dual basis, as follows:

if f is constant:

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

and if f is balanced:

DEUTSCH PROBLEM-RANDOMIZED SOLUTION

Given a function $f : \{0,1\} \to \{0,1\}$, as a **black box**, the task is to determine whether f is constant or balanced.
To solve the problem:

In classical computing 2 calls of f are needed.

In quantum computing 1 call of f is sufficient.

Quantum algorithm presented below solves the problem with probability $\frac{1}{2}$ in such a way that we know whether the answer is correct. Since

$$U_f : (\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)|0\rangle) \to \frac{1}{\sqrt{2}}(|0, f(0)\rangle + |1, f(1)\rangle),$$

the result can be written, in the standard and dual basis, as follows:

if f is constant:

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

and if f is balanced:

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

Therefore if f is constant:

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

and if f is balanced:

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

Therefore if f is constant:

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

and if f is balanced:

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

If the measurement of the second qubit in the dual bases provides 0 we have lost all information about f.

Therefore if f is constant:

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

and if f is balanced:

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

If the measurement of the second qubit in the dual bases provides 0 we have lost all information about f. Otherwise, the measurement of the first qubit yields the correct result.

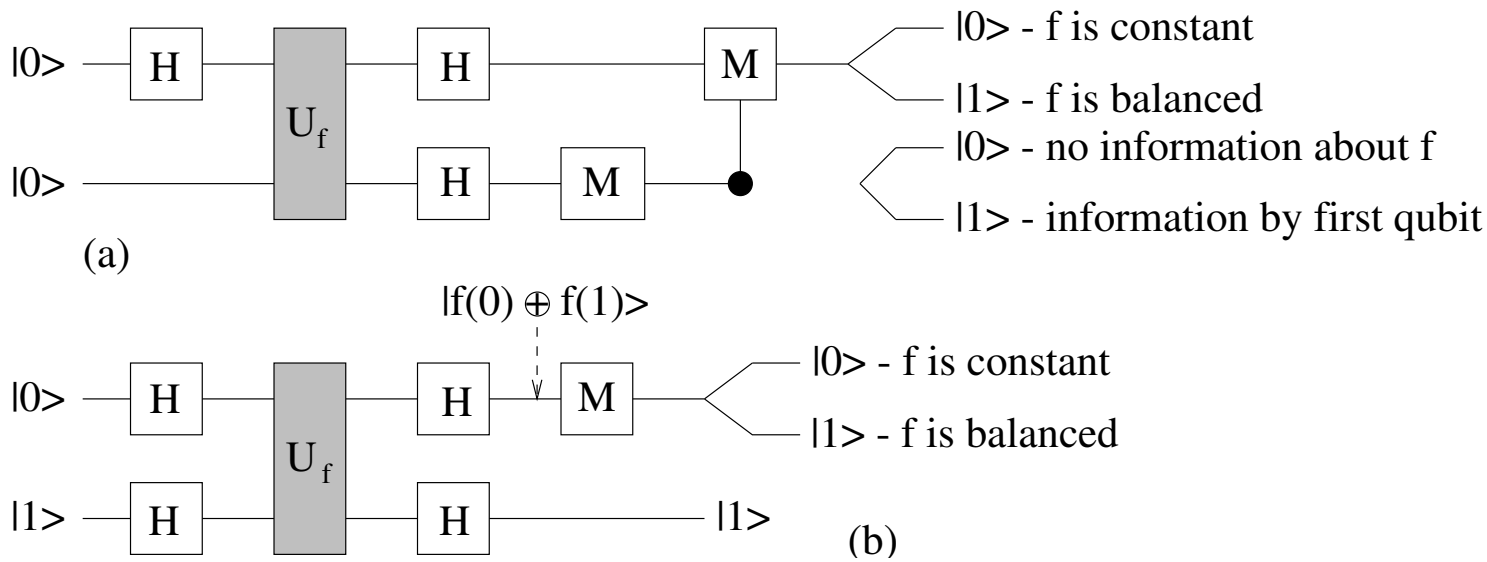
Therefore if f is constant:

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

and if f is balanced:

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

If the measurement of the second qubit in the dual bases provides 0 we have lost all information about f. Otherwise, the measurement of the first qubit yields the correct result.
The corresponding circuit is shown in the following Figure (a).



DETERMINISTIC SOLUTION

DETERMINISTIC SOLUTION

Let Hadamard transforms are applied on both registers in the initial state $|0,1\rangle$

DETERMINISTIC SOLUTION

Let Hadamard transforms are applied on both registers in the initial state $|0,1\rangle$ and then the unitary U_f is applied we get

$$|0\rangle|1\rangle \quad \stackrel{H_2}{\rightarrow} \quad \frac{1}{2}(|0\rangle + |1\rangle)(|0\rangle - |1\rangle)$$

DETERMINISTIC SOLUTION

Let Hadamard transforms are applied on both registers in the initial state $|0,1\rangle$ and then the unitary U_f is applied we get

$$\begin{aligned} |0\rangle|1\rangle &\xrightarrow{H_2} \frac{1}{2}(|0\rangle + |1\rangle)(|0\rangle - |1\rangle) \\ &= \frac{1}{2}(|0\rangle(|0\rangle - |1\rangle) + |1\rangle(|0\rangle - |1\rangle)) \end{aligned}$$

DETERMINISTIC SOLUTION

Let Hadamard transforms are applied on both registers in the initial state $|0,1\rangle$ and then the unitary U_f is applied we get

$$\begin{aligned}
|0\rangle|1\rangle &\xrightarrow{H_2} \frac{1}{2}(|0\rangle + |1\rangle)(|0\rangle - |1\rangle) \\
&= \frac{1}{2}(|0\rangle(|0\rangle - |1\rangle) + |1\rangle(|0\rangle - |1\rangle)) \\
&\xrightarrow{U_f} \frac{1}{2}(|0\rangle(|0 \oplus f(0)\rangle - |1 \oplus f(0)\rangle) + |1\rangle(|0 \oplus f(1)\rangle - |1 \oplus f(1)\rangle))
\end{aligned}$$

DETERMINISTIC SOLUTION

Let Hadamard transforms are applied on both registers in the initial state $|0,1\rangle$ and then the unitary U_f is applied we get

$$\begin{aligned}
|0\rangle|1\rangle &\xrightarrow{H_2} \frac{1}{2}(|0\rangle + |1\rangle)(|0\rangle - |1\rangle) \\
&= \frac{1}{2}(|0\rangle(|0\rangle - |1\rangle) + |1\rangle(|0\rangle - |1\rangle)) \\
&\xrightarrow{U_f} \frac{1}{2}(|0\rangle(|0 \oplus f(0)\rangle - |1 \oplus f(0)\rangle) + |1\rangle(|0 \oplus f(1)\rangle - |1 \oplus f(1)\rangle)) \\
&= \frac{1}{2}(\sum_{x=0}^{1}(-1)^{f(x)}|x\rangle)(|0\rangle - |1\rangle)
\end{aligned}$$

DETERMINISTIC SOLUTION

Let Hadamard transforms are applied on both registers in the initial state $|0,1\rangle$ and then the unitary U_f is applied we get

$$\begin{aligned}
|0\rangle|1\rangle &\xrightarrow{H_2} \frac{1}{2}(|0\rangle+|1\rangle)(|0\rangle-|1\rangle) \\
&= \frac{1}{2}(|0\rangle(|0\rangle-|1\rangle)+|1\rangle(|0\rangle-|1\rangle)) \\
&\xrightarrow{U_f} \frac{1}{2}(|0\rangle(|0\oplus f(0)\rangle-|1\oplus f(0)\rangle)+|1\rangle(|0\oplus f(1)\rangle-|1\oplus f(1)\rangle)) \\
&= \frac{1}{2}(\sum_{x=0}^{1}(-1)^{f(x)}|x\rangle)(|0\rangle-|1\rangle) \\
&= \frac{1}{2}(-1)^{f(0)}(|0\rangle+(-1)^{f(0)\oplus f(1)}|1\rangle)(|0\rangle-|1\rangle). \qquad (1)
\end{aligned}$$

From the right side in (1), the two possibilities for f to be constant lead to the left sides in (2) and (3) and two possibilities for f to be balanced lead to the left sides in (4) and (5):

$$\frac{1}{2}(|0\rangle + |1\rangle)(|0\rangle - |1\rangle) = |0'\rangle|1'\rangle \text{ if } f(0) = 0; \quad (2)$$

$$\frac{1}{2}(|0\rangle + |1\rangle)(|1\rangle - |0\rangle) = -|0'\rangle|1'\rangle \text{ if } f(0) = 1; \quad (3)$$

$$\frac{1}{2}(|0\rangle - |1\rangle)(|0\rangle - |1\rangle) = |1'\rangle|1'\rangle \text{ if } f(0) = 0; \quad (4)$$

$$\frac{1}{2}(|0\rangle - |1\rangle)(|1\rangle - |0\rangle) = -|1'\rangle|1'\rangle \text{ if } f(0) = 1. \quad (5)$$

From the right side in (1), the two possibilities for f to be constant lead to the left sides in (2) and (3) and two possibilities for f to be balanced lead to the left sides in (4) and (5):

$$\frac{1}{2}(|0\rangle + |1\rangle)(|0\rangle - |1\rangle) = |0'\rangle|1'\rangle \text{ if } f(0) = 0; \quad (2)$$

$$\frac{1}{2}(|0\rangle + |1\rangle)(|1\rangle - |0\rangle) = -|0'\rangle|1'\rangle \text{ if } f(0) = 1; \quad (3)$$

$$\frac{1}{2}(|0\rangle - |1\rangle)(|0\rangle - |1\rangle) = |1'\rangle|1'\rangle \text{ if } f(0) = 0; \quad (4)$$

$$\frac{1}{2}(|0\rangle - |1\rangle)(|1\rangle - |0\rangle) = -|1'\rangle|1'\rangle \text{ if } f(0) = 1. \quad (5)$$

By measuring the first bit, with respect to the dual basis, we can immediately see whether f is constant or balanced.

EVEN-ODD PROBLEM

EVEN-ODD PROBLEM

A function $f : \{0,1\}^2 \leftrightarrow \{0,1\}$ is called even (odd) if the range of f has even (odd) number of ones.

EVEN-ODD PROBLEM

A function $f : \{0,1\}^2 \leftrightarrow \{0,1\}$ is called even (odd) if the range of f has even (odd) number of ones.

Classically, given such a function f as an oracle, one needs 4 calls of f to determine whether f is even or odd.

EVEN-ODD PROBLEM

A function $f : \{0,1\}^2 \leftrightarrow \{0,1\}$ is called even (odd) if the range of f has even (odd) number of ones.

Classically, given such a function f as an oracle, one needs 4 calls of f to determine whether f is even or odd.

Quantumly, it holds

$$(H \otimes H)V_f(I \otimes H)V_f(H \otimes H)|00\rangle = \begin{cases} \frac{1}{\sqrt{2}}(\pm|00\rangle + |01\rangle) \text{ if f is even} \\ \frac{1}{\sqrt{2}}(\pm|10\rangle + |01\rangle) \text{ if f is odd} \end{cases}$$

EVEN-ODD PROBLEM

A function $f : \{0,1\}^2 \leftrightarrow \{0,1\}$ is called even (odd) if the range of f has even (odd) number of ones.

Classically, given such a function f as an oracle, one needs 4 calls of f to determine whether f is even or odd.

Quantumly, it holds

$$(H \otimes H)V_f(I \otimes H)V_f(H \otimes H)|00\rangle = \begin{cases} \frac{1}{\sqrt{2}}(\pm|00\rangle + |01\rangle) \text{ if f is even} \\ \frac{1}{\sqrt{2}}(\pm|10\rangle + |01\rangle) \text{ if f is odd} \end{cases}$$

and therefore using only two quantum calls of f (of V_f), the problem is transformed into the problem to distinguish two non-orthogonal quantum states.

EVEN-ODD PROBLEM

A function $f : \{0,1\}^2 \leftrightarrow \{0,1\}$ is called even (odd) if the range of f has even (odd) number of ones.

Classically, given such a function f as an oracle, one needs 4 calls of f to determine whether f is even or odd.

Quantumly, it holds

$$(H \otimes H)V_f(I \otimes H)V_f(H \otimes H)|00\rangle = \begin{cases} \frac{1}{\sqrt{2}}(\pm|00\rangle + |01\rangle) \text{ if f is even} \\ \frac{1}{\sqrt{2}}(\pm|10\rangle + |01\rangle) \text{ if f is odd} \end{cases}$$

and therefore using only two quantum calls of f (of V_f), the problem is transformed into the problem to distinguish two non-orthogonal quantum states.

Unfortunately, there is no projection measurement that can faithfully distinguish such non-orthogonal states.

DEUTSCH-JOZSA PROBLEM

DEUTSCH-JOZSA PROBLEM

Given a function $f : \{0,1\}^n \to \{0,1\}$, as a black box, that is (promised to be) balanced or constant.

DEUTSCH-JOZSA PROBLEM

Given a function $f : \{0,1\}^n \to \{0,1\}$, as a black box, that is (promised to be) balanced or constant. Decide which property f has.

DEUTSCH-JOZSA PROBLEM

Given a function $f : \{0,1\}^n \to \{0,1\}$, as a black box, that is (promised to be) balanced or constant. Decide which property f has.

Classical deterministic computers need, in the worst case, exponential time to solve the problem.

DEUTSCH-JOZSA PROBLEM

Given a function $f : \{0,1\}^n \to \{0,1\}$, as a black box, that is (promised to be) balanced or constant. Decide which property f has.

Classical deterministic computers need, in the worst case, exponential time to solve the problem. Surprisingly, there is a quantum algorithm to solve the problem by applying f only once.

DEUTSCH-JOZSA PROBLEM

Given a function $f : \{0,1\}^n \to \{0,1\}$, as a black box, that is (promised to be) balanced or constant. Decide which property f has.

Classical deterministic computers need, in the worst case, exponential time to solve the problem. Surprisingly, there is a quantum algorithm to solve the problem by applying f only once.

Let us consider one quantum register with n qubits and apply the Hadamard transformation H_n to the first register.

DEUTSCH-JOZSA PROBLEM

Given a function $f : \{0,1\}^n \to \{0,1\}$, as a black box, that is (promised to be) balanced or constant. Decide which property f has.

Classical deterministic computers need, in the worst case, exponential time to solve the problem. Surprisingly, there is a quantum algorithm to solve the problem by applying f only once.

Let us consider one quantum register with n qubits and apply the Hadamard transformation H_n to the first register. This yields

$$|0^{(n)}\rangle \xrightarrow{H_n} |\phi\rangle = \frac{1}{\sqrt{2^n}} \sum_{i=0}^{2^n-1} |i\rangle.$$

DEUTSCH-JOZSA PROBLEM

Given a function $f : \{0,1\}^n \to \{0,1\}$, as a black box, that is (promised to be) balanced or constant. Decide which property f has.

Classical deterministic computers need, in the worst case, exponential time to solve the problem. Surprisingly, there is a quantum algorithm to solve the problem by applying f only once.

Let us consider one quantum register with n qubits and apply the Hadamard transformation H_n to the first register. This yields

$$|0^{(n)}\rangle \xrightarrow{H_n} |\phi\rangle = \frac{1}{\sqrt{2^n}} \sum_{i=0}^{2^n-1} |i\rangle.$$

By applying the transformation V_f on the first register we get

$$V_f|\phi\rangle = \frac{1}{\sqrt{2^n}} \sum_{i=0}^{2^n-1} (-1)^{f(i)}|i\rangle = |\phi_1\rangle.$$

DEUTSCH-JOZSA PROBLEM

Given a function $f : \{0,1\}^n \to \{0,1\}$, as a black box, that is (promised to be) balanced or constant. Decide which property f has.

Classical deterministic computers need, in the worst case, exponential time to solve the problem. Surprisingly, there is a quantum algorithm to solve the problem by applying f only once.

Let us consider one quantum register with n qubits and apply the Hadamard transformation H_n to the first register. This yields

$$|0^{(n)}\rangle \xrightarrow{H_n} |\phi\rangle = \frac{1}{\sqrt{2^n}} \sum_{i=0}^{2^n-1} |i\rangle.$$

By applying the transformation V_f on the first register we get

$$V_f|\phi\rangle = \frac{1}{\sqrt{2^n}} \sum_{i=0}^{2^n-1} (-1)^{f(i)}|i\rangle = |\phi_1\rangle.$$

What has been achieved by these operations? The values of f were transferred to the amplitudes!!!

This can be utilized, through the power of quantum superposition and a proper observable, as follows.

This can be utilized, through the power of quantum superposition and a proper observable, as follows.

Let us consider the observable $\mathcal{D} = \{E_a, E_b\}$, where E_a is the one-dimensional subspace spanned by the vector

$$|\psi_a\rangle = \frac{1}{\sqrt{2^n}} \sum_{i=0}^{2^n-1} |i\rangle,$$

This can be utilized, through the power of quantum superposition and a proper observable, as follows.

Let us consider the observable $\mathcal{D} = \{E_a, E_b\}$, where E_a is the one-dimensional subspace spanned by the vector

$$|\psi_a\rangle = \frac{1}{\sqrt{2^n}} \sum_{i=0}^{2^n-1} |i\rangle,$$

and $E_b = (E_a)^{\perp}$.

This can be utilized, through the power of quantum superposition and a proper observable, as follows.

Let us consider the observable $\mathcal{D} = \{E_a, E_b\}$, where E_a is the one-dimensional subspace spanned by the vector

$$|\psi_a\rangle = \frac{1}{\sqrt{2^n}} \sum_{i=0}^{2^n-1} |i\rangle,$$

and $E_b = (E_a)^{\perp}$. The projection of $|\phi_1\rangle$ into E_a and E_b has the form

$$|\phi_1\rangle = \alpha|\psi_a\rangle + \beta|\psi_b\rangle \quad \text{with} \quad |\alpha|^2 + |\beta|^2 = 1,$$

where $|\psi_b\rangle$ is a vector in E_b such that $|\psi_a\rangle \perp |\psi_b\rangle$.

This can be utilized, through the power of quantum superposition and a proper observable, as follows.

Let us consider the observable $\mathcal{D} = \{E_a, E_b\}$, where E_a is the one-dimensional subspace spanned by the vector

$$|\psi_a\rangle = \frac{1}{\sqrt{2^n}} \sum_{i=0}^{2^n-1} |i\rangle,$$

and $E_b = (E_a)^{\perp}$. The projection of $|\phi_1\rangle$ into E_a and E_b has the form

$$|\phi_1\rangle = \alpha|\psi_a\rangle + \beta|\psi_b\rangle \quad \text{with} \quad |\alpha|^2 + |\beta|^2 = 1,$$

where $|\psi_b\rangle$ is a vector in E_b such that $|\psi_a\rangle \perp |\psi_b\rangle$. A measurement by $\mathcal{D}$ provides "the value a or b" with probability $|\alpha|^2$ or $|\beta|^2$.

This can be utilized, through the power of quantum superposition and a proper observable, as follows.

Let us consider the observable $\mathcal{D} = \{E_a, E_b\}$, where E_a is the one-dimensional subspace spanned by the vector

$$|\psi_a\rangle = \frac{1}{\sqrt{2^n}} \sum_{i=0}^{2^n-1} |i\rangle,$$

and $E_b = (E_a)^{\perp}$. The projection of $|\phi_1\rangle$ into E_a and E_b has the form

$$|\phi_1\rangle = \alpha|\psi_a\rangle + \beta|\psi_b\rangle \quad \text{with} \quad |\alpha|^2 + |\beta|^2 = 1,$$

where $|\psi_b\rangle$ is a vector in E_b such that $|\psi_a\rangle \perp |\psi_b\rangle$. A measurement by $\mathcal{D}$ provides "the value a or b" with probability $|\alpha|^2$ or $|\beta|^2$.
It is easy to determine α in

$$|\phi_1\rangle = \alpha|\psi_a\rangle + \beta|\psi_b\rangle \quad \text{with} \quad |\alpha|^2 + |\beta|^2 = 1,$$

using the projection of $|\phi_1\rangle$ onto E_a

This can be utilized, through the power of quantum superposition and a proper observable, as follows.

Let us consider the observable $\mathcal{D} = \{E_a, E_b\}$, where E_a is the one-dimensional subspace spanned by the vector

$$|\psi_a\rangle = \frac{1}{\sqrt{2^n}} \sum_{i=0}^{2^n-1} |i\rangle,$$

and $E_b = (E_a)^{\perp}$. The projection of $|\phi_1\rangle$ into E_a and E_b has the form

$$|\phi_1\rangle = \alpha|\psi_a\rangle + \beta|\psi_b\rangle \quad \text{with} \quad |\alpha|^2 + |\beta|^2 = 1,$$

where $|\psi_b\rangle$ is a vector in E_b such that $|\psi_a\rangle \perp |\psi_b\rangle$. A measurement by $\mathcal{D}$ provides "the value a or b" with probability $|\alpha|^2$ or $|\beta|^2$.
It is easy to determine α in

$$|\phi_1\rangle = \alpha|\psi_a\rangle + \beta|\psi_b\rangle \quad \text{with} \quad |\alpha|^2 + |\beta|^2 = 1,$$

using the projection of $|\phi_1\rangle$ onto E_a by the computation

$$\alpha = \langle\psi_a|\phi_1\rangle.$$

Indeed

$$\alpha \quad =$$

Indeed

$$\alpha \quad = \quad \langle \psi_a | \phi_1 \rangle$$

Indeed

$$\alpha \quad = \quad \langle \psi_a | \phi_1 \rangle = \left(\frac{1}{\sqrt{2^n}} \sum_{i=0}^{2^n-1} \langle i | \right) \left(\frac{1}{\sqrt{2^n}} \sum_{j=0}^{2^n-1} (-1)^{f(j)} | j \rangle \right)$$

Indeed

$$\alpha = \langle\psi_a|\phi_1\rangle = \left(\frac{1}{\sqrt{2^n}}\sum_{i=0}^{2^n-1}\langle i|\right)\left(\frac{1}{\sqrt{2^n}}\sum_{j=0}^{2^n-1}(-1)^{f(j)}|j\rangle\right)$$

$$= \frac{1}{2^n}\sum_{i=0}^{2^n-1}\sum_{j=0}^{2^n-1}(-1)^{f(j)}\langle i|j\rangle$$

Indeed

$$\alpha = \langle\psi_a|\phi_1\rangle = \left(\frac{1}{\sqrt{2^n}}\sum_{i=0}^{2^n-1}\langle i|\right)\left(\frac{1}{\sqrt{2^n}}\sum_{j=0}^{2^n-1}(-1)^{f(j)}|j\rangle\right)$$

$$= \frac{1}{2^n}\sum_{i=0}^{2^n-1}\sum_{j=0}^{2^n-1}(-1)^{f(j)}\langle i|j\rangle = \frac{1}{2^n}\sum_{i=0}^{2^n-1}(-1)^{f(i)},$$

Indeed

$$\begin{aligned}\alpha &= \langle\psi_a|\phi_1\rangle = \left(\frac{1}{\sqrt{2^n}}\sum_{i=0}^{2^n-1}\langle i|\right)\left(\frac{1}{\sqrt{2^n}}\sum_{j=0}^{2^n-1}(-1)^{f(j)}|j\rangle\right)\\ &= \frac{1}{2^n}\sum_{i=0}^{2^n-1}\sum_{j=0}^{2^n-1}(-1)^{f(j)}\langle i|j\rangle = \frac{1}{2^n}\sum_{i=0}^{2^n-1}(-1)^{f(i)},\end{aligned}$$

because $\langle i|j\rangle = 1$ if and only if $i = j$ and 0 otherwise.

Indeed

$$\begin{aligned}\alpha &= \langle\psi_a|\phi_1\rangle = \left(\frac{1}{\sqrt{2^n}}\sum_{i=0}^{2^n-1}\langle i|\right)\left(\frac{1}{\sqrt{2^n}}\sum_{j=0}^{2^n-1}(-1)^{f(j)}|j\rangle\right)\\ &= \frac{1}{2^n}\sum_{i=0}^{2^n-1}\sum_{j=0}^{2^n-1}(-1)^{f(j)}\langle i|j\rangle = \frac{1}{2^n}\sum_{i=0}^{2^n-1}(-1)^{f(i)},\end{aligned}$$

because $\langle i|j\rangle = 1$ if and only if $i = j$ and 0 otherwise.

If f is balanced, then the sum for α contains the same number of 1s and -1s and therefore $\alpha = 0$. A measurement of $|\phi_1\rangle$, with respect to $\mathcal{D}$ therefore provides, for sure, the outcome b.

Indeed

$$\alpha = \langle\psi_a|\phi_1\rangle = \left(\frac{1}{\sqrt{2^n}}\sum_{i=0}^{2^n-1}\langle i|\right)\left(\frac{1}{\sqrt{2^n}}\sum_{j=0}^{2^n-1}(-1)^{f(j)}|j\rangle\right)$$

$$= \frac{1}{2^n}\sum_{i=0}^{2^n-1}\sum_{j=0}^{2^n-1}(-1)^{f(j)}\langle i|j\rangle = \frac{1}{2^n}\sum_{i=0}^{2^n-1}(-1)^{f(i)},$$

because $\langle i|j\rangle = 1$ if and only if $i = j$ and 0 otherwise.

If f is balanced, then the sum for α contains the same number of 1s and -1s and therefore $\alpha = 0$. A measurement of $|\phi_1\rangle$, with respect to $\mathcal{D}$ therefore provides, for sure, the outcome b.

If f is constant, then either $\alpha = 1$ or $\alpha = -1$ and therefore the measurement of $|\phi_1\rangle$ with respect to $\mathcal{D}$ always gives the outcome a.

Indeed

$$\alpha = \langle\psi_a|\phi_1\rangle = \left(\frac{1}{\sqrt{2^n}}\sum_{i=0}^{2^n-1}\langle i|\right)\left(\frac{1}{\sqrt{2^n}}\sum_{j=0}^{2^n-1}(-1)^{f(j)}|j\rangle\right)$$

$$= \frac{1}{2^n}\sum_{i=0}^{2^n-1}\sum_{j=0}^{2^n-1}(-1)^{f(j)}\langle i|j\rangle = \frac{1}{2^n}\sum_{i=0}^{2^n-1}(-1)^{f(i)},$$

because $\langle i|j\rangle = 1$ if and only if $i = j$ and 0 otherwise.

If f is balanced, then the sum for α contains the same number of 1s and -1s and therefore $\alpha = 0$. A measurement of $|\phi_1\rangle$, with respect to $\mathcal{D}$ therefore provides, for sure, the outcome b.

If f is constant, then either $\alpha = 1$ or $\alpha = -1$ and therefore the measurement of $|\phi_1\rangle$ with respect to $\mathcal{D}$ always gives the outcome a.

A single measurement of $|\phi_1\rangle$, with respect to $\mathcal{D}$, therefore provides the solution of the problem with probability 1.

SECOND SOLUTION

SECOND SOLUTION

If the Hadamard transformation is applied to the state $|\phi_1\rangle$ we get the state

$$|\phi_2\rangle =$$

SECOND SOLUTION

If the Hadamard transformation is applied to the state $|\phi_1\rangle$ we get the state

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

SECOND SOLUTION

If the Hadamard transformation is applied to the state $|\phi_1\rangle$ we get the state

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

Case 1: f is constant.

SECOND SOLUTION

If the Hadamard transformation is applied to the state $|\phi_1\rangle$ we get the state

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

Case 1: f is constant. Then

$$\sum_{i=0}^{2^n-1} (-1)^{u \cdot i} = \begin{cases} 0 & \text{if } u \neq 0 \\ 2^n & \text{if } u = 0 \end{cases}.$$

SECOND SOLUTION

If the Hadamard transformation is applied to the state $|\phi_1\rangle$ we get the state

$$|\phi_2\rangle = \frac{1}{\sqrt{2^n}} \sum_{i=0}^{2^n-1} (-1)^{f(i)} \frac{1}{\sqrt{2^n}} \sum_{u=0}^{2^n-1} (-1)^{u \cdot i} |u\rangle = \frac{1}{2^n} \sum_{u=0}^{2^n-1} \left(\sum_{i=0}^{2^n-1} (-1)^{u \cdot i} (-1)^{f(i)} \right) |u\rangle.$$

Case 1: f is constant. Then

$$\sum_{i=0}^{2^n-1} (-1)^{u \cdot i} = \begin{cases} 0 & \text{if } u \neq 0 \\ 2^n & \text{if } u = 0 \end{cases}.$$

One measurement of the register therefore provides $u = 0$ with probability 1.

SECOND SOLUTION

If the Hadamard transformation is applied to the state $|\phi_1\rangle$ we get the state

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

Case 1: f is constant. Then

$$\sum_{i=0}^{2^n-1} (-1)^{u \cdot i} = \begin{cases} 0 & \text{if } u \neq 0 \\ 2^n & \text{if } u = 0 \end{cases}.$$

One measurement of the register therefore provides $u = 0$ with probability 1.

Case 2: f is balanced.

SECOND SOLUTION

If the Hadamard transformation is applied to the state $|\phi_1\rangle$ we get the state

$$|\phi_2\rangle = \frac{1}{\sqrt{2^n}} \sum_{i=0}^{2^n-1} (-1)^{f(i)} \frac{1}{\sqrt{2^n}} \sum_{u=0}^{2^n-1} (-1)^{u \cdot i} |u\rangle = \frac{1}{2^n} \sum_{u=0}^{2^n-1} \left(\sum_{i=0}^{2^n-1} (-1)^{u \cdot i} (-1)^{f(i)} \right) |u\rangle.$$

Case 1: f is constant. Then

$$\sum_{i=0}^{2^n-1} (-1)^{u \cdot i} = \begin{cases} 0 & \text{if } u \neq 0 \\ 2^n & \text{if } u = 0 \end{cases}.$$

One measurement of the register therefore provides $u = 0$ with probability 1.

Case 2: f is balanced. In such a case

$$\sum_{i=0}^{2^n-1} (-1)^{u \cdot i} (-1)^{f(i)} = 0 \text{ if and only if } u = 0.$$

SECOND SOLUTION

If the Hadamard transformation is applied to the state $|\phi_1\rangle$ we get the state

$$|\phi_2\rangle = \frac{1}{\sqrt{2^n}} \sum_{i=0}^{2^n-1} (-1)^{f(i)} \frac{1}{\sqrt{2^n}} \sum_{u=0}^{2^n-1} (-1)^{u \cdot i} |u\rangle = \frac{1}{2^n} \sum_{u=0}^{2^n-1} \left(\sum_{i=0}^{2^n-1} (-1)^{u \cdot i} (-1)^{f(i)} \right) |u\rangle.$$

Case 1: f is constant. Then

$$\sum_{i=0}^{2^n-1} (-1)^{u \cdot i} = \begin{cases} 0 & \text{if } u \neq 0 \\ 2^n & \text{if } u = 0 \end{cases}.$$

One measurement of the register therefore provides $u = 0$ with probability 1.

Case 2: f is balanced. In such a case

$$\sum_{i=0}^{2^n-1} (-1)^{u \cdot i} (-1)^{f(i)} = 0 \text{ if and only if } u = 0.$$

One measurement therefore shows whether f is balanced or not.

DJ-PROBLEM - CLASSICAL RANDOM SOLUTION

DJ-PROBLEM - CLASSICAL RANDOM SOLUTION

It is easy to show that though deterministic algorithms to solve the Deutsch-Jozsa problem for $n = 2^k$ require $2^{k-1} + 1$ queries in the worst case, there are probabilistic algorithms to solve this problem relatively fast, if we are willing to tolerate some error.

DJ-PROBLEM - CLASSICAL RANDOM SOLUTION

It is easy to show that though deterministic algorithms to solve the Deutsch-Jozsa problem for $n = 2^k$ require $2^{k-1} + 1$ queries in the worst case, there are probabilistic algorithms to solve this problem relatively fast, if we are willing to tolerate some error.

Indeed, a randomized algorithm can solve the Deutsch-Jozsa problem with probability of error at most $\frac{1}{3}$ with only two queries.

DJ-PROBLEM - CLASSICAL RANDOM SOLUTION

It is easy to show that though deterministic algorithms to solve the Deutsch-Jozsa problem for $n = 2^k$ require $2^{k-1} + 1$ queries in the worst case, there are probabilistic algorithms to solve this problem relatively fast, if we are willing to tolerate some error.

Indeed, a randomized algorithm can solve the Deutsch-Jozsa problem with probability of error at most $\frac{1}{3}$ with only two queries.

The probability of error can be reduced to less than $\frac{1}{2^k}$ with only $k + 1$ queries.

DJ-PROBLEM - CLASSICAL RANDOM SOLUTION

It is easy to show that though deterministic algorithms to solve the Deutsch-Jozsa problem for $n = 2^k$ require $2^{k-1} + 1$ queries in the worst case, there are probabilistic algorithms to solve this problem relatively fast, if we are willing to tolerate some error.

Indeed, a randomized algorithm can solve the Deutsch-Jozsa problem with probability of error at most $\frac{1}{3}$ with only two queries.

The probability of error can be reduced to less than $\frac{1}{2^k}$ with only $k + 1$ queries.

Therefore, in spite of the fact that there is an exponential gap between deterministic classical and exact quantum query complexity, the gap between randomized classical complexity and quantum query complexity is in this case constant in the case of constant error.

SIMON's PROBLEM

SIMON's PROBLEM

Simon has discovered a simple problem with polynomial expected time quantum algorithm, but with no polynomial time randomized algorithm.

SIMON's PROBLEM

Simon has discovered a simple problem with polynomial expected time quantum algorithm, but with no polynomial time randomized algorithm.

SIMON's PROBLEM

Simon has discovered a simple problem with polynomial expected time quantum algorithm, but with no polynomial time randomized algorithm.

Let $f : \{0,1\}^n \to \{0,1\}^n$ be a function such that either f is one-to-one or f is two-to-one and there exists a single $0 \neq s \in \{0,1\}^n$ such that

$$\forall x \neq x'(f(x) = f(x') \Leftrightarrow x' = x \oplus s).$$

SIMON's PROBLEM

Simon has discovered a simple problem with polynomial expected time quantum algorithm, but with no polynomial time randomized algorithm.

Let $f : \{0,1\}^n \to \{0,1\}^n$ be a function such that either f is one-to-one or f is two-to-one and there exists a single $0 \neq s \in \{0,1\}^n$ such that

$$\forall x \neq x'(f(x) = f(x') \Leftrightarrow x' = x \oplus s).$$

The task is to determine which of the above conditions holds for f and, in the second case, to determine also s.

To solve the problem two registers are used, both with n qubits, and the initial states $|0^{(n)}\rangle$, and (expected) $\mathcal{O}(n)$ repetitions of the following version of the so-called Hadamard-twice scheme:

To solve the problem two registers are used, both with n qubits, and the initial states $|0^{(n)}\rangle$, and (expected) $\mathcal{O}(n)$ repetitions of the following version of the so-called Hadamard-twice scheme:

1. *Apply the Hadamard transformation on the first register, with the initial value* $|0^{(n)}\rangle$*, to produce the superposition* $\frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} |x, 0^{(n)}\rangle$.

To solve the problem two registers are used, both with n qubits, and the initial states $|0^{(n)}\rangle$, and (expected) $\mathcal{O}(n)$ repetitions of the following version of the so-called Hadamard-twice scheme:

1. *Apply the Hadamard transformation on the first register, with the initial value* $|0^{(n)}\rangle$*, to produce the superposition* $\frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} |x, 0^{(n)}\rangle$.
2. *Apply* U_f *to compute* $|\psi\rangle = \frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} |x, f(x)\rangle$.

To solve the problem two registers are used, both with n qubits, and the initial states $|0^{(n)}\rangle$, and (expected) $\mathcal{O}(n)$ repetitions of the following version of the so-called Hadamard-twice scheme:

1. *Apply the Hadamard transformation on the first register, with the initial value* $|0^{(n)}\rangle$*, to produce the superposition* $\frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} |x, 0^{(n)}\rangle$.
2. *Apply* U_f *to compute* $|\psi\rangle = \frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} |x, f(x)\rangle$.
3. *Apply Hadamard transformation on the first register to get*

$$\frac{1}{2^n} \sum_{x,y\in\{0,1\}^n} (-1)^{x\cdot y} |y, f(x)\rangle.$$

To solve the problem two registers are used, both with n qubits, and the initial states $|0^{(n)}\rangle$, and (expected) $\mathcal{O}(n)$ repetitions of the following version of the so-called Hadamard-twice scheme:

1. *Apply the Hadamard transformation on the first register, with the initial value* $|0^{(n)}\rangle$*, to produce the superposition* $\frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} |x, 0^{(n)}\rangle$.
2. *Apply* U_f *to compute* $|\psi\rangle = \frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} |x, f(x)\rangle$.
3. *Apply Hadamard transformation on the first register to get*

$$\frac{1}{2^n} \sum_{x,y\in\{0,1\}^n} (-1)^{x\cdot y} |y, f(x)\rangle.$$

4. *Observe the resulting state to get a pair* $(y, f(x))$.

To solve the problem two registers are used, both with n qubits, and the initial states $|0^{(n)}\rangle$, and (expected) $\mathcal{O}(n)$ repetitions of the following version of the so-called Hadamard-twice scheme:

1. *Apply the Hadamard transformation on the first register, with the initial value* $|0^{(n)}\rangle$*, to produce the superposition* $\frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} |x, 0^{(n)}\rangle$.
2. *Apply* U_f *to compute* $|\psi\rangle = \frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} |x, f(x)\rangle$.
3. *Apply Hadamard transformation on the first register to get*

$$\frac{1}{2^n} \sum_{x,y\in\{0,1\}^n} (-1)^{x\cdot y} |y, f(x)\rangle.$$

4. *Observe the resulting state to get a pair* $(y, f(x))$.

Case 1: f is one-to-one. After performing the first three steps of the above procedure all possible states $|y, f(x)\rangle$ in the superposition are distinct and the absolute value of their amplitudes is the same, namely 2^{-n}.

To solve the problem two registers are used, both with n qubits, and the initial states $|0^{(n)}\rangle$, and (expected) $\mathcal{O}(n)$ repetitions of the following version of the so-called Hadamard-twice scheme:

1. *Apply the Hadamard transformation on the first register, with the initial value* $|0^{(n)}\rangle$*, to produce the superposition* $\frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} |x, 0^{(n)}\rangle$.
2. *Apply* U_f *to compute* $|\psi\rangle = \frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} |x, f(x)\rangle$.
3. *Apply Hadamard transformation on the first register to get*

$$\frac{1}{2^n} \sum_{x,y\in\{0,1\}^n} (-1)^{x\cdot y} |y, f(x)\rangle.$$

4. *Observe the resulting state to get a pair* $(y, f(x))$.

Case 1: f is one-to-one. After performing the first three steps of the above procedure all possible states $|y, f(x)\rangle$ in the superposition are distinct and the absolute value of their amplitudes is the same, namely 2^{-n}.

$n-1$ independent applications of the scheme *Hadamard-twice* therefore produce $n-1$ pairs $(y_1, f(x_1)), \ldots, (y_{n-1}, f(x_{n-1}))$, distributed uniformly and independently over all pairs $(y, f(x))$.

To solve the problem two registers are used, both with n qubits, and the initial states $|0^{(n)}\rangle$, and (expected) $\mathcal{O}(n)$ repetitions of the following version of the so-called Hadamard-twice scheme:

1. *Apply the Hadamard transformation on the first register, with the initial value* $|0^{(n)}\rangle$*, to produce the superposition* $\frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} |x, 0^{(n)}\rangle$.
2. *Apply* U_f *to compute* $|\psi\rangle = \frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} |x, f(x)\rangle$.
3. *Apply Hadamard transformation on the first register to get*

$$\frac{1}{2^n} \sum_{x,y\in\{0,1\}^n} (-1)^{x\cdot y} |y, f(x)\rangle.$$

4. *Observe the resulting state to get a pair* $(y, f(x))$.

Case 1: f is one-to-one. After performing the first three steps of the above procedure all possible states $|y, f(x)\rangle$ in the superposition are distinct and the absolute value of their amplitudes is the same, namely 2^{-n}.

$n-1$ independent applications of the scheme *Hadamard-twice* therefore produce $n-1$ pairs $(y_1, f(x_1)), \ldots, (y_{n-1}, f(x_{n-1}))$, distributed uniformly and independently over all pairs $(y, f(x))$.

Case 2: There is some $s \neq 0^{(n)}$ such that

$$\forall x \neq x'((f(x) = f(x') \Leftrightarrow x' = x \oplus s).$$

Case 2: There is some $s \neq 0^{(n)}$ such that

$$\forall x \neq x'((f(x) = f(x') \Leftrightarrow x' = x \oplus s).$$

In such a case for each y and x the states $|y, f(x)\rangle$ and $|y, f(x \oplus s)\rangle$ are identical.

Case 2: There is some $s \neq 0^{(n)}$ such that

$$\forall x \neq x'((f(x) = f(x') \Leftrightarrow x' = x \oplus s).$$

In such a case for each y and x the states $|y, f(x)\rangle$ and $|y, f(x \oplus s)\rangle$ are identical. Their total amplitude $\alpha(x, y)$ has therefore the value

$$\alpha(x, y) = 2^{-n}((-1)^{x \cdot y} + (-1)^{(x \oplus s) \cdot y}).$$

Case 2: There is some $s \neq 0^{(n)}$ such that

$$\forall x \neq x'((f(x) = f(x') \Leftrightarrow x' = x \oplus s).$$

In such a case for each y and x the states $|y, f(x)\rangle$ and $|y, f(x \oplus s)\rangle$ are identical. Their total amplitude $\alpha(x, y)$ has therefore the value

$$\alpha(x, y) = 2^{-n}((-1)^{x \cdot y} + (-1)^{(x \oplus s) \cdot y}).$$

If $y \cdot s \equiv 0 \bmod 2$, then $x \cdot y \equiv (x \oplus s) \cdot y \bmod 2$ and therefore $|\alpha(x, y)| = 2^{-n+1}$; otherwise $\alpha(x, y) = 0$.

Case 2: There is some $s \neq 0^{(n)}$ such that

$$\forall x \neq x'((f(x) = f(x') \Leftrightarrow x' = x \oplus s).$$

In such a case for each y and x the states $|y, f(x)\rangle$ and $|y, f(x \oplus s)\rangle$ are identical. Their total amplitude $\alpha(x, y)$ has therefore the value

$$\alpha(x, y) = 2^{-n}((-1)^{x \cdot y} + (-1)^{(x \oplus s) \cdot y}).$$

If $y \cdot s \equiv 0 \bmod 2$, then $x \cdot y \equiv (x \oplus s) \cdot y \bmod 2$ and therefore $|\alpha(x, y)| = 2^{-n+1}$; otherwise $\alpha(x, y) = 0$. n independent applications of the scheme *Hadamard-twice* therefore yield $n - 1$ independent pairs

$$(y_1, f(x_1)), \ldots, (y_{n-1}, f(x_{n-1})) \text{ such that } y_i \cdot s \equiv 0 \pmod 2,$$

for all $1 \leq i \leq n - 1$.

Case 2: There is some $s \neq 0^{(n)}$ such that

$$\forall x \neq x'((f(x) = f(x') \Leftrightarrow x' = x \oplus s).$$

In such a case for each y and x the states $|y, f(x)\rangle$ and $|y, f(x \oplus s)\rangle$ are identical. Their total amplitude $\alpha(x, y)$ has therefore the value

$$\alpha(x, y) = 2^{-n}((-1)^{x \cdot y} + (-1)^{(x \oplus s) \cdot y}).$$

If $y \cdot s \equiv 0 \bmod 2$, then $x \cdot y \equiv (x \oplus s) \cdot y \bmod 2$ and therefore $|\alpha(x, y)| = 2^{-n+1}$; otherwise $\alpha(x, y) = 0$. n independent applications of the scheme *Hadamard-twice* therefore yield $n - 1$ independent pairs

$$(y_1, f(x_1)), \ldots, (y_{n-1}, f(x_{n-1})) \text{ such that } y_i \cdot s \equiv 0 \pmod 2,$$

for all $1 \leq i \leq n - 1$.

In both cases, after $n - 1$ repetitions of the scheme *Hadamard-twice*, $n - 1$ vectors $y_i, 1 \leq i \leq n - 1$, are obtained.

Case 2: There is some $s \neq 0^{(n)}$ such that

$$\forall x \neq x'((f(x) = f(x') \Leftrightarrow x' = x \oplus s).$$

In such a case for each y and x the states $|y, f(x)\rangle$ and $|y, f(x \oplus s)\rangle$ are identical. Their total amplitude $\alpha(x, y)$ has therefore the value

$$\alpha(x, y) = 2^{-n}((-1)^{x \cdot y} + (-1)^{(x \oplus s) \cdot y}).$$

If $y \cdot s \equiv 0 \bmod 2$, then $x \cdot y \equiv (x \oplus s) \cdot y \bmod 2$ and therefore $|\alpha(x, y)| = 2^{-n+1}$; otherwise $\alpha(x, y) = 0$. n independent applications of the scheme *Hadamard-twice* therefore yield $n - 1$ independent pairs

$$(y_1, f(x_1)), \ldots, (y_{n-1}, f(x_{n-1})) \text{ such that } y_i \cdot s \equiv 0 \pmod 2,$$

for all $1 \leq i \leq n - 1$.

In both cases, after $n - 1$ repetitions of the scheme *Hadamard-twice*, $n - 1$ vectors $y_i, 1 \leq i \leq n - 1$, are obtained.

If these vectors are linearly independent, then the system of $n - 1$ linear equations in $\mathbf{Z}_2$,

$$y_i \cdot s \equiv 0 \pmod n$$

can be solved to obtain s.

In Case 2, if f is two-to-one, s obtained in such a way is the one to be found.

In Case 2, if f is two-to-one, s obtained in such a way is the one to be found.

In Case 1, s obtained in such a way is a random string.

In Case 2, if f is two-to-one, s obtained in such a way is the one to be found.

In Case 1, s obtained in such a way is a random string.

To distinguish these two cases, it is enough to compute $f(0)$ and $f(s)$.

In Case 2, if f is two-to-one, s obtained in such a way is the one to be found.

In Case 1, s obtained in such a way is a random string.

To distinguish these two cases, it is enough to compute $f(0)$ and $f(s)$.

If $f(0) \neq f(s)$, then f is one-to-one. If the vectors obtained by the scheme

Hadamard-twice are not linearly independent, then the whole process has to be repeated.

LOWER BOUND

LOWER BOUND

We show that each classical algorithm needs to perform $\Omega(\sqrt{2^n})$ queries to solve Simon's problem.

LOWER BOUND

We show that each classical algorithm needs to perform $\Omega(\sqrt{2^n})$ queries to solve Simon's problem.

Indeed, let us assume that f is a randomly chosen function satisfying requirements of the Simon's problem.

LOWER BOUND

We show that each classical algorithm needs to perform $\Omega(\sqrt{2^n})$ queries to solve Simon's problem.

Indeed, let us assume that f is a randomly chosen function satisfying requirements of the Simon's problem.

If k f-queries are performed then the number of potential s is decreased at most by $\frac{k(k-1)}{2}$ possibilities.

LOWER BOUND

We show that each classical algorithm needs to perform $\Omega(\sqrt{2^n})$ queries to solve Simon's problem.

Indeed, let us assume that f is a randomly chosen function satisfying requirements of the Simon's problem.

If k f-queries are performed then the number of potential s is decreased at most by $\frac{k(k-1)}{2}$ possibilities.

In total there are 2^n potential s.

LOWER BOUND

We show that each classical algorithm needs to perform $\Omega(\sqrt{2^n})$ queries to solve Simon's problem.

Indeed, let us assume that f is a randomly chosen function satisfying requirements of the Simon's problem.

If k f-queries are performed then the number of potential s is decreased at most by $\frac{k(k-1)}{2}$ possibilities.

In total there are 2^n potential s.

Hence at least in half of the cases any classical algorithm needs to perform $\Omega(\sqrt{2^n})$ f-queries.

COMPUTATIONAL POWER of ENTANGLEMENT

COMPUTATIONAL POWER of ENTANGLEMENT

As illustrated in the following examples, in some cases there is a clever way to make use of quantum entanglement to compute efficiently some global properties of a function.

COMPUTATIONAL POWER of ENTANGLEMENT

As illustrated in the following examples, in some cases there is a clever way to make use of quantum entanglement to compute efficiently some global properties of a function.

Let a function $f : \{1, \ldots, n\} \rightarrow \{0, 1\}$ be given as a black box.

To determine f classically, n calls of f are needed—

COMPUTATIONAL POWER of ENTANGLEMENT

As illustrated in the following examples, in some cases there is a clever way to make use of quantum entanglement to compute efficiently some global properties of a function.

Let a function $f : \{1, \ldots, n\} \rightarrow \{0, 1\}$ be given as a black box.

To determine f classically, n calls of f are needed—to get the string $w_f = f(1)f(2)\ldots f(n)$.

COMPUTATIONAL POWER of ENTANGLEMENT

As illustrated in the following examples, in some cases there is a clever way to make use of quantum entanglement to compute efficiently some global properties of a function.

Let a function $f : \{1, \ldots, n\} \rightarrow \{0, 1\}$ be given as a black box.

To determine f classically, n calls of f are needed—to get the string $w_f = f(1)f(2)\ldots f(n)$.

Quantumly, this can be done, with probability greater than 0.95, using $\frac{n}{2} + \sqrt{n}$ quantum calls of f.

COMPUTATIONAL POWER of ENTANGLEMENT

As illustrated in the following examples, in some cases there is a clever way to make use of quantum entanglement to compute efficiently some global properties of a function.

Let a function $f : \{1, \ldots, n\} \rightarrow \{0, 1\}$ be given as a black box.

To determine f classically, n calls of f are needed—to get the string $w_f = f(1)f(2)\ldots f(n)$.

Quantumly, this can be done, with probability greater than 0.95, using $\frac{n}{2} + \sqrt{n}$ quantum calls of f.

Indeed, on the base of equality

$$|w_f\rangle = H_n \frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} (-1)^{x \cdot w_f} |x\rangle \tag{6}$$

COMPUTATIONAL POWER of ENTANGLEMENT

As illustrated in the following examples, in some cases there is a clever way to make use of quantum entanglement to compute efficiently some global properties of a function.

Let a function $f : \{1, \ldots, n\} \rightarrow \{0, 1\}$ be given as a black box.

To determine f classically, n calls of f are needed—to get the string $w_f = f(1)f(2)\ldots f(n)$.

Quantumly, this can be done, with probability greater than 0.95, using $\frac{n}{2} + \sqrt{n}$ quantum calls of f.

Indeed, on the base of equality

$$|w_f\rangle = H_n \frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} (-1)^{x \cdot w_f} |x\rangle \tag{6}$$

in order to compute $x \cdot w_f$ one needs $hw(x)$ calls of f, where $hw(x)$ is the Hamming weight of x—the number of 1' in x.

COMPUTATIONAL POWER of ENTANGLEMENT

As illustrated in the following examples, in some cases there is a clever way to make use of quantum entanglement to compute efficiently some global properties of a function.

Let a function $f : \{1, \ldots, n\} \rightarrow \{0, 1\}$ be given as a black box.

To determine f classically, n calls of f are needed—to get the string $w_f = f(1)f(2)\ldots f(n)$.

Quantumly, this can be done, with probability greater than 0.95, using $\frac{n}{2} + \sqrt{n}$ quantum calls of f.

Indeed, on the base of equality

$$|w_f\rangle = H_n \frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} (-1)^{x \cdot w_f} |x\rangle \tag{6}$$

in order to compute $x \cdot w_f$ one needs $hw(x)$ calls of f, where $hw(x)$ is the Hamming weight of x—the number of 1' in x.

The basic trick is to compute the sum in (6) but only for x such that $hw(x) \leq k$,

Let F_k be such a function that for $x \in \{0,1\}^n$,

$$F_k(x) = \begin{cases} x \cdot w_f \text{ if } hw(x) \leq k \\ 0; \text{ otherwise} \end{cases}$$

Let F_k be such a function that for $x \in \{0,1\}^n$,

$$F_k(x) = \begin{cases} x \cdot w_f \text{ if } hw(x) \leq k \\ 0; \text{ otherwise} \end{cases}$$

In such a case

$$V_{F_k}|x\rangle = \begin{cases} (-1)^{x \cdot w_f}|x\rangle, \text{ if } hw(x) \leq k \\ |x\rangle; \text{ otherwise} \end{cases}$$

Let F_k be such a function that for $x \in \{0,1\}^n$,

$$F_k(x) = \begin{cases} x \cdot w_f \text{ if } hw(x) \leq k \\ 0; \text{ otherwise} \end{cases}$$

In such a case

$$V_{F_k}|x\rangle = \begin{cases} (-1)^{x \cdot w_f}|x\rangle, \text{ if } hw(x) \leq k \\ |x\rangle; \text{ otherwise} \end{cases}$$

Therefore if V_{F_k} is applied to the (initial) state

$$|\psi_k\rangle = \frac{1}{\sqrt{M_k}} \sum_{x \in \{0,1\}^n}^{hw(x) \leq k} |x\rangle,$$

where $M_k = \sum_{i=0}^{k} \binom{n}{i}$,

Let F_k be such a function that for $x \in \{0,1\}^n$,

$$F_k(x) = \begin{cases} x \cdot w_f \text{ if } hw(x) \leq k \\ 0; \text{ otherwise} \end{cases}$$

In such a case

$$V_{F_k}|x\rangle = \begin{cases} (-1)^{x \cdot w_f}|x\rangle, \text{ if } hw(x) \leq k \\ |x\rangle; \text{ otherwise} \end{cases}$$

Therefore if V_{F_k} is applied to the (initial) state

$$|\psi_k\rangle = \frac{1}{\sqrt{M_k}} \sum_{x \in \{0,1\}^n}^{hw(x) \leq k} |x\rangle,$$

where $M_k = \sum_{i=0}^{k} \binom{n}{i}$, then

$$|\psi'_k\rangle = V_{F_k}|\psi_k\rangle = \frac{1}{\sqrt{M_k}} \sum_{x \in \{0,1\}^n}^{hw(x) \leq k} (-1)^{x \cdot w_f}|x\rangle.$$

Let F_k be such a function that for $x \in \{0,1\}^n$,

$$F_k(x) = \begin{cases} x \cdot w_f \text{ if } hw(x) \leq k \\ 0; \text{ otherwise} \end{cases}$$

In such a case

$$V_{F_k}|x\rangle = \begin{cases} (-1)^{x \cdot w_f}|x\rangle, \text{ if } hw(x) \leq k \\ |x\rangle; \text{ otherwise} \end{cases}$$

Therefore if V_{F_k} is applied to the (initial) state

$$|\psi_k\rangle = \frac{1}{\sqrt{M_k}} \sum_{x \in \{0,1\}^n}^{hw(x) \leq k} |x\rangle,$$

where $M_k = \sum_{i=0}^{k} \binom{n}{i}$, then

$$|\psi'_k\rangle = V_{F_k}|\psi_k\rangle = \frac{1}{\sqrt{M_k}} \sum_{x \in \{0,1\}^n}^{hw(x) \leq k} (-1)^{x \cdot w_f}|x\rangle.$$

In order to compute $|\psi'_k\rangle$, at most k calls of f are needed.

Let F_k be such a function that for $x \in \{0,1\}^n$,

$$F_k(x) = \begin{cases} x \cdot w_f \text{ if } hw(x) \leq k \\ 0; \text{ otherwise} \end{cases}$$

In such a case

$$V_{F_k}|x\rangle = \begin{cases} (-1)^{x \cdot w_f}|x\rangle, \text{ if } hw(x) \leq k \\ |x\rangle; \text{ otherwise} \end{cases}$$

Therefore if V_{F_k} is applied to the (initial) state

$$|\psi_k\rangle = \frac{1}{\sqrt{M_k}} \sum_{x \in \{0,1\}^n}^{hw(x) \leq k} |x\rangle,$$

where $M_k = \sum_{i=0}^{k} \binom{n}{i}$, then

$$|\psi'_k\rangle = V_{F_k}|\psi_k\rangle = \frac{1}{\sqrt{M_k}} \sum_{x \in \{0,1\}^n}^{hw(x) \leq k} (-1)^{x \cdot w_f}|x\rangle.$$

In order to compute $|\psi'_k\rangle$, at most k calls of f are needed. Let us now measure all n qubits of $|\psi''_k\rangle = H_n|\psi'_k\rangle$.

The probability that this way we get w_f is

$$Pr(|\psi''_k\rangle \text{ yields at measurement } w_f) = |\langle w_f|\psi''_k\rangle|^2 = \frac{M_k}{2^n} = \frac{1}{2^n}\sum_{i=1}^{k}\binom{n}{i}$$

and, as one can easily calculate, this probability is more than 0.95 if $k = \frac{n}{2} + \sqrt{n}$.

DE-QUANTIZATION of DEUTSCH PROBLEM

DE-QUANTIZATION of DEUTSCH PROBLEM

Surprisingly, quantum algorithms for Deutsch problem can be de-quantised as follows:

DE-QUANTIZATION of DEUTSCH PROBLEM

Surprisingly, quantum algorithms for Deutsch problem can be de-quantised as follows:

For a given $f : \{0,1\} \to \{0,1\}$ we define an oraculum mapping

$$C_f(a+bi) = (-1)^{0 \oplus f(0)} a + (-1)^{1 \oplus f(1)} bi$$

For the four possible functions f we get the following four functions C_f:

$$\begin{array}{lll} C_{00}(x) = x^* & \text{if} & f(0)=0, f(1)=0 \\ C_{01}(x) = x & \text{if} & f(0)=0, f(1)=1 \\ C_{10}(x) = -x & \text{if} & f(0)=1, f(1)=0 \\ C_{11}(x) = -x^* & \text{if} & f(0)=1, f(1)=1 \end{array}$$

DE-QUANTIZATION of DEUTSCH PROBLEM

Surprisingly, quantum algorithms for Deutsch problem can be de-quantised as follows:

For a given $f : \{0,1\} \to \{0,1\}$ we define an oraculum mapping

$$C_f(a+bi) = (-1)^{0 \oplus f(0)} a + (-1)^{1 \oplus f(1)} bi$$

For the four possible functions f we get the following four functions C_f:

$$\begin{array}{lll} C_{00}(x) = x^* & \text{if} & f(0) = 0, f(1) = 0 \\ C_{01}(x) = x & \text{if} & f(0) = 0, f(1) = 1 \\ C_{10}(x) = -x & \text{if} & f(0) = 1, f(1) = 0 \\ C_{11}(x) = -x^* & \text{if} & f(0) = 1, f(1) = 1 \end{array}$$

The Deutsch problem can now be formulated as follows: A function is chosen secretly from the set of functions $\{C_{00}, C_{01}, C_{10}, C_{11}\}$ and the task is to determine, with a single query, which type of the function it is - balanced or constant.

Algorithm Given f, calculate $(i-1)C_f(1+i)$. If the outcome is real, then the function chosen is balanced; otherwise it is constant.

Algorithm Given f, calculate $(i-1)C_f(1+i)$. If the outcome is real, then the function chosen is balanced; otherwise it is constant.

Correctness:

$$\begin{aligned}
(i-1)C_{00}(1+i) &= (i-1)(1-i) = 2i \\
(i-1)C_{01}(1+i) &= (i-1)(1+i) = -2 \\
(i-1)C_{10}(1+i) &= (i-1)(-1-i) = 2 \\
(i-1)C_{11}(1+i) &= (i-1)(1-i) = -2i
\end{aligned}$$