4. QUANTUM COMPUTING

Jozef Gruska

Faculty of Informatics
Brno
Czech Republic

April 23, 2007

4. QUANTUM CIRCUITS

Quantum circuits are the most easy to deal with model of quantum computations.

Several simple quantum gates form elementary building blocks from which any quantum circuit can be built.

PART I - MOTIVATION

MOTTO I.

Progress in science is often done by pessimists.
Progress in technology is always done by optimists.

MOTTO II.

Progress in science is often done by pessimists.
Progress in technology is always done by knowledgeable and experienced optimists.

TWO STORIES TO REMEMBER

- The proposal to build Collosus, the first electronic computer for cryptanalysis purposes, was during the 2WW rejected by a committee of prominent specialists as impossible to make, in spite of the fact that British cryptanalysts needed it badly to crack communication between Hitler and his generals.
- Collosus was then built by an ingenious optimist, Tommy Flowers, within 10 months in a Post office laboratory, and worked from the beginning successfully to break Lorenz cipher, starting January 1944.
- The key point was that Flowers realized that velvets were reliable provided they were never switched on and off. (Of course, nobody believed him.)
- The idea that 30m long ENIAC with 19000 vacuum tubes could work looked also crazy, for scientists, but it worked.

BASIC OBSERVATIONS – I

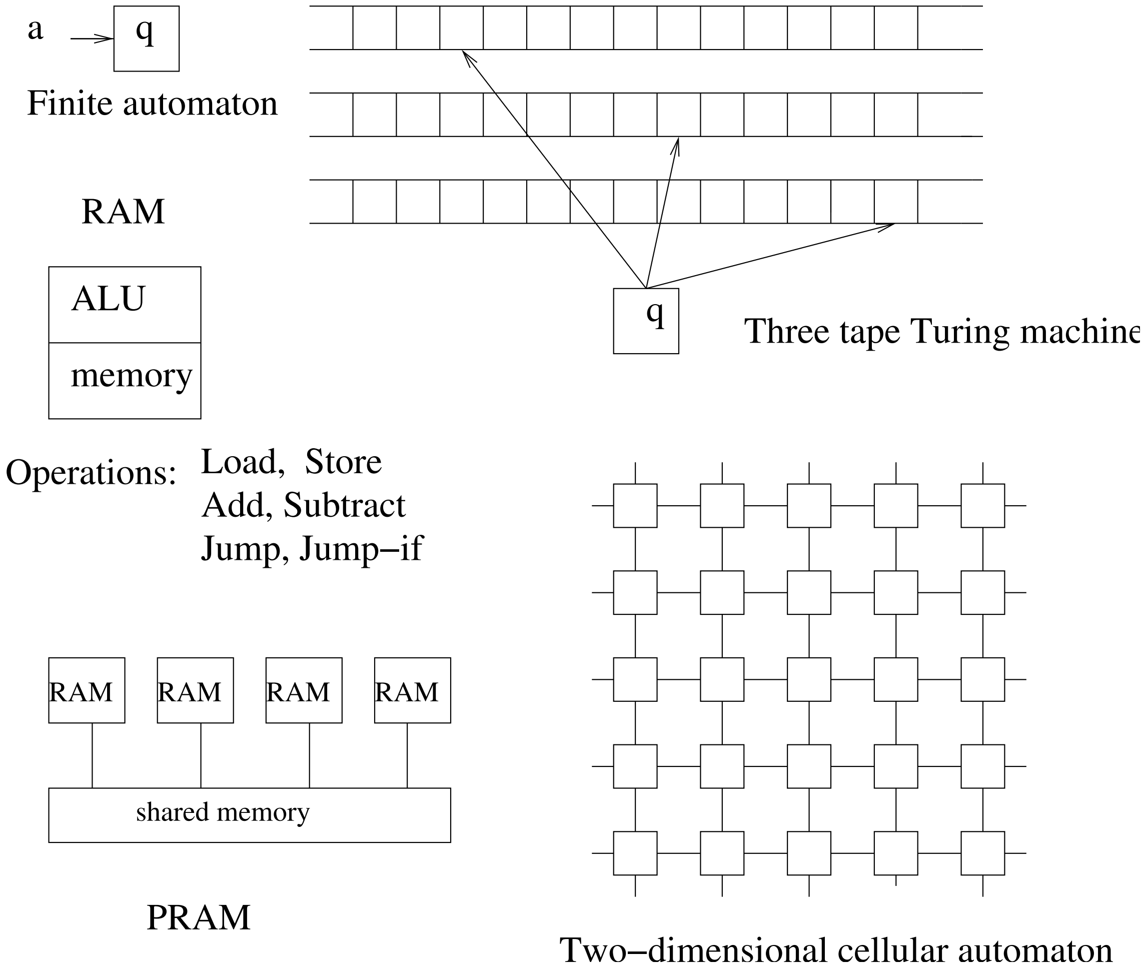
Observation: An apparently small observation of a scientists or an experience of an engineer can turn a field upside down and “create a superstar from a sleeping beauty”.

Conclusion: It is very, very important to search for primitives and for new and new primitives - even in the areas one can hardly expect them.

MODELS of UNIVERSAL COMPUTERS

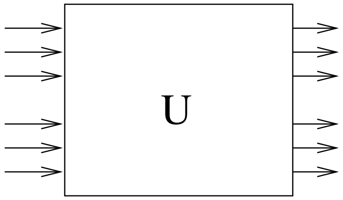
- Classical models: circuits, Turing machines, cellular automata, RAM a PRAM
- Quantum models
  - (Unitary operations based) Quantum Turing Machines
  - (Unitary operations based) Quantum Circuits
  - Quantum cellular automata ????
  - Measurements based quantum circuits
  - Measurements based quantum Turing machines
- Emerging idea: Classically controlled quantum computation (automata).

MAIN MODELS of AUTOMATA



QUANTUM GATES — SIMPLE EXAMPLES

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 (rotation) gates are represented by matrices

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

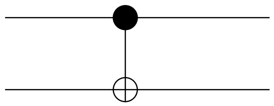
Example The following one parameter set of **rotation gates** (represented by matrices) is also often used:

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

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

QUANTUM GATES → UNITARY MATRICES

Unitary matrix for XOR-gate (CNOT-gate)

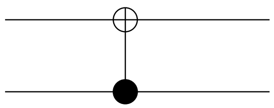


has the form

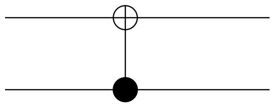
$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}.$$

What is the unitary matrix representing the "inverse XOR gate"?

h



h



What is the unitary matrix representing the “inverse XOR gate”?

It has the form

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

SOLVING SCHRÖDINGER EQUATION

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

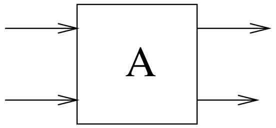
$$U(t) = e^{-\frac{i}{\hbar}Ht} = \sum_{k=0}^{\infty}\frac{(-\frac{i\pi}{2})^k V^k t^k}{k!} = I + \frac{1}{2}\sum_{k=1}^{\infty}\frac{(-\pi it)^k}{k!}V$$

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

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

QUANTUM GATES $\rightarrow$ UNITARY MATRICES

For a quantum gate

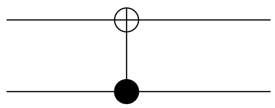


A **is such a unitary matrix**

$$\begin{array}{c|cccc} & \mathbf{00} & \mathbf{01} & \mathbf{10} & \mathbf{11} \\ \hline \mathbf{00} & \cdot & \cdot & \cdot & \cdot \\ \mathbf{01} & \cdot & \cdot & \cdot & \cdot \\ \mathbf{10} & \cdot & \cdot & \cdot & \cdot \\ \mathbf{11} & \cdot & \cdot & \cdot & \cdot \end{array}$$

that the entry for column ij **and row** kl **is the probability amplitude that the basis state** $|ij\rangle$ **transfers to the basis state** $|kl\rangle$**.**

Example For the inverse XOR gate



the unitary matrix has the form

	00	01	10	11
00	1	0	0	0
01	0	0	0	1
10	0	0	1	0
11	0	1	0	0

FROM GATES to UNITARY MATRICES

In general, if a quantum gate has n inputs and outputs then for the corresponding unitary matrix the entry

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

and

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

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

UNITARY MATRICES versus BASES

Each unitary operator has 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

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

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

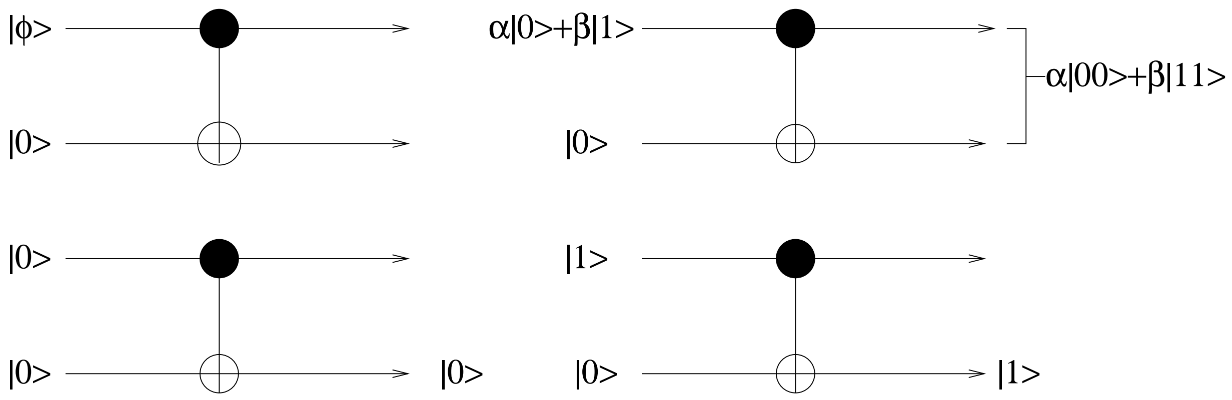
$$|\Phi^+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \qquad |\Phi^-\rangle = \frac{1}{\sqrt{2}}(|00\rangle - |11\rangle)$$

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

has the form

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

XOR GATE as a REAL WIRE



As already mentioned, of central importance for quantum computing is the XOR gate (see leftmost and topmost figure).

Observe that if the target qubit has the input $|0\rangle$, then this gate can be used to copy qubits $|0\rangle$ and $|1\rangle$ from the control qubit.

At the same time the gate in Figure a can be seen as a classical wire because it cannot carry on a superposition of the states.

A QUANTUM EVOLUTION STEP

A quantum evolution step consists formally of a quantum state (vector) multiplication by a unitary operator. That is

$$A|\phi\rangle = |\psi\rangle$$

For example,

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{pmatrix} \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{pmatrix} = \begin{pmatrix} a_{11}b_1 + a_{12}b_2 + a_{13}b_3 + a_{14}b_4 \\ a_{21}b_1 + a_{22}b_2 + a_{23}b_3 + a_{24}b_4 \\ a_{31}b_1 + a_{32}b_2 + a_{33}b_3 + a_{34}b_4 \\ a_{41}b_1 + a_{42}b_2 + a_{43}b_3 + a_{44}b_4 \end{pmatrix}.$$

A better insight into such a process can be obtained using different notation at which it is assumed that all rows and columns are labeled by the states of the standard basis of H_4.

$$\begin{pmatrix} a_{00,00} & a_{00,01} & a_{00,10} & a_{00,11} \\ a_{01,00} & a_{01,01} & a_{01,10} & a_{01,11} \\ a_{10,00} & a_{10,01} & a_{10,10} & a_{10,11} \\ a_{11,00} & a_{11,01} & a_{11,10} & a_{11,11} \end{pmatrix} \begin{pmatrix} b_{00} \\ b_{01} \\ b_{10} \\ b_{11} \end{pmatrix} = \begin{pmatrix} b_{00}a_{00,00} + b_{01}a_{00,01} + b_{10}a_{00,10} + b_{11}a_{00,11} \\ b_{00}a_{01,00} + b_{01}a_{01,01} + b_{10}a_{01,10} + b_{11}a_{01,11} \\ b_{00}a_{10,00} + b_{01}a_{10,01} + b_{10}a_{10,10} + b_{11}a_{10,11} \\ b_{00}a_{11,00} + b_{01}a_{11,01} + b_{10}a_{11,10} + b_{11}a_{11,11} \end{pmatrix} = \begin{pmatrix} d_{00} \\ d_{01} \\ d_{10} \\ d_{11} \end{pmatrix}.$$

QUANTUM CIRCUITS — SIMPLE EXAMPLES

defi A **quantum (Boolean) circuit** is a collection of quantum gates acyclically connected (by "quantum wires"). defi

A relation between a quantum circuit and the corresponding unitary matrix is far from being very transparent even for simple circuits and some experience is needed to get proper feelings in this respect.

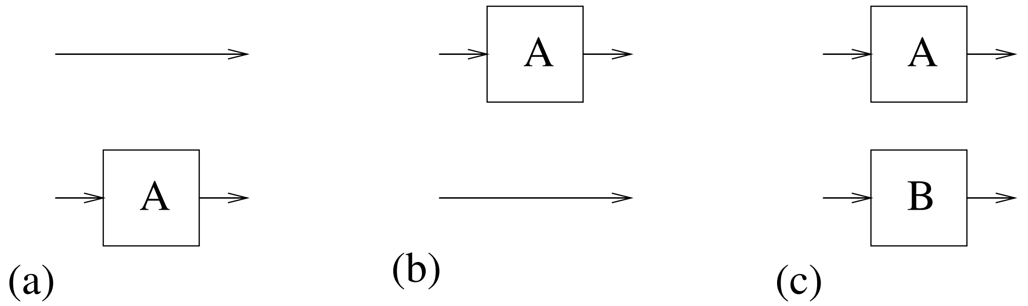


Figure 1: Elementary networks I

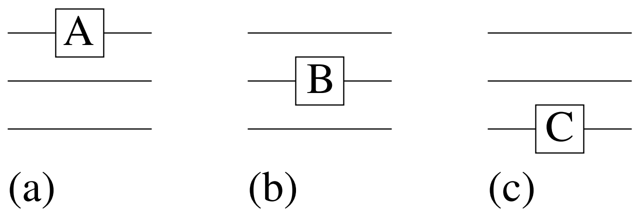


Figure 2: Elementary networks II

INVERSE XOR GATE CIRCUIT

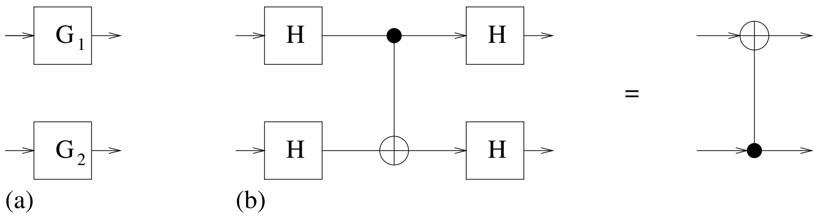


Figure 3: An implementation of the inverse of the XOR gate.

The processing in the network on the left side of the identity in Figure 3b for the input $|0\rangle|1\rangle$ can be depicted as follows:

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

EXAMPLE

There are various generalizations of XOR gates:

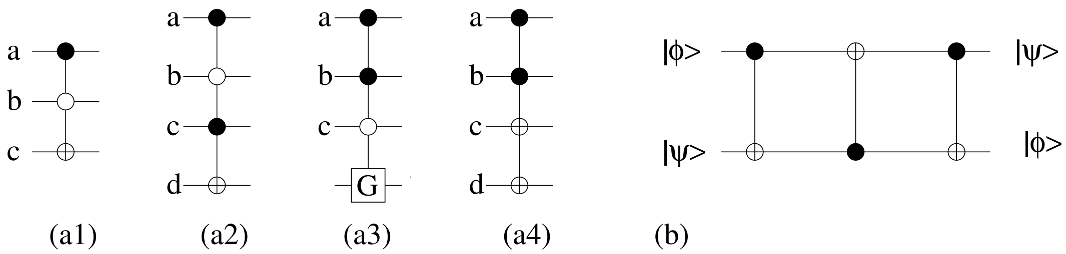


Figure 4: Generalized XOR gate notations and a quantum circuit to flip the qubits

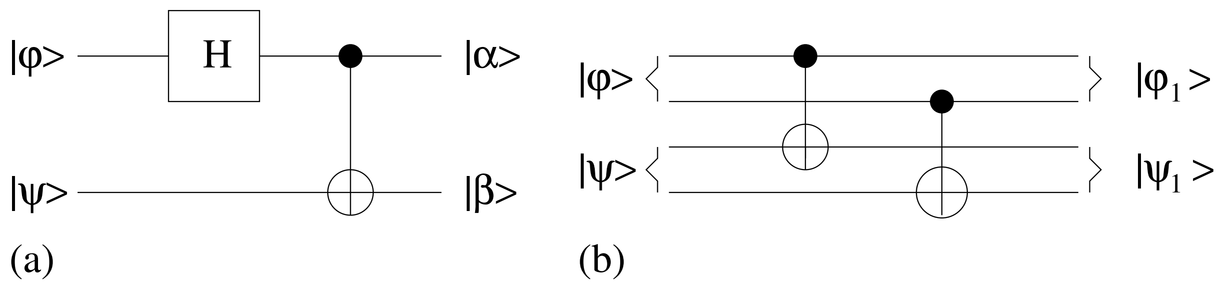
The circuit in Figure b realizes flipping of qubits.

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:

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

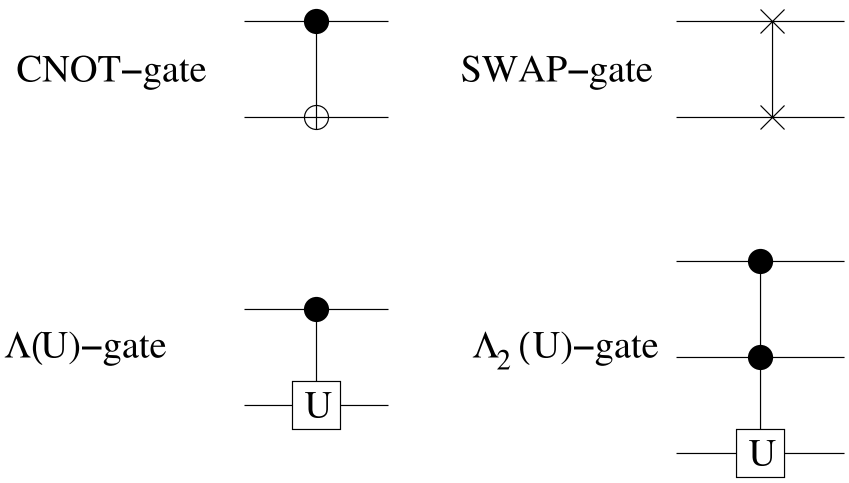
BELL STATE CIRCUITS



The circuit in Figure a produces all four Bell states for all possibilities $|\phi\rangle, |\psi\rangle \in \{|0\rangle, |1\rangle\}$.

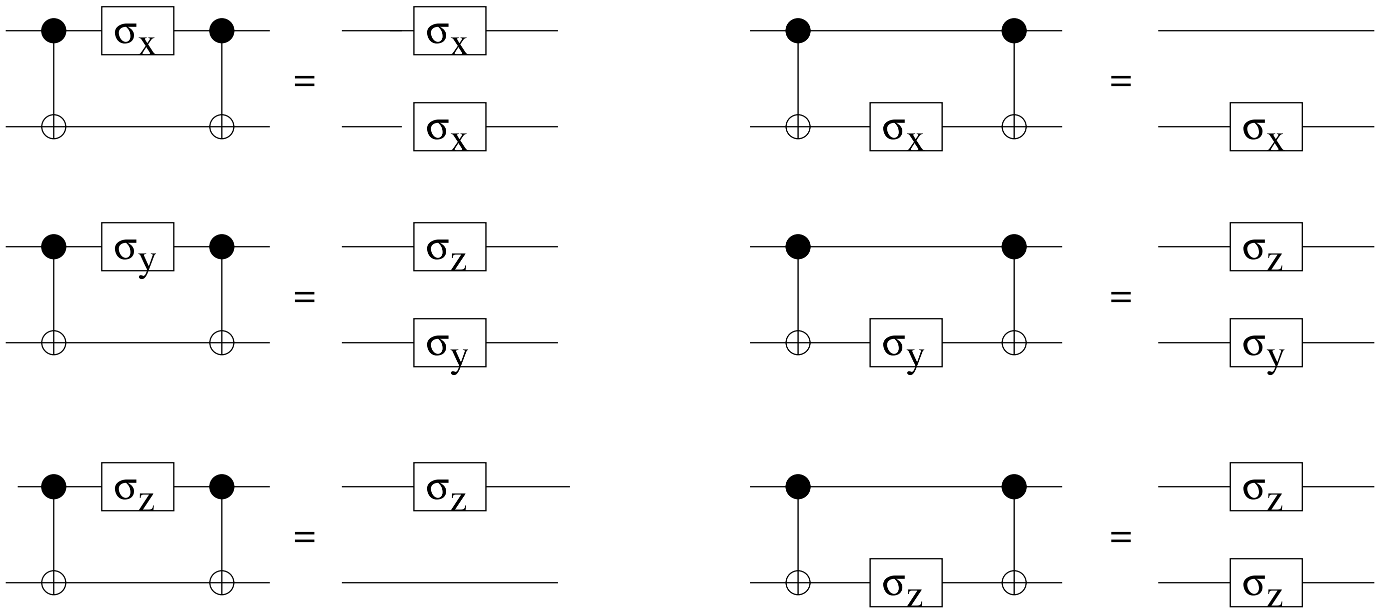
The circuit in Figure b performs one-to-one mapping of Bell states into Bell states.

GRAPHICAL REPRESENTATION of some BASIC GATES

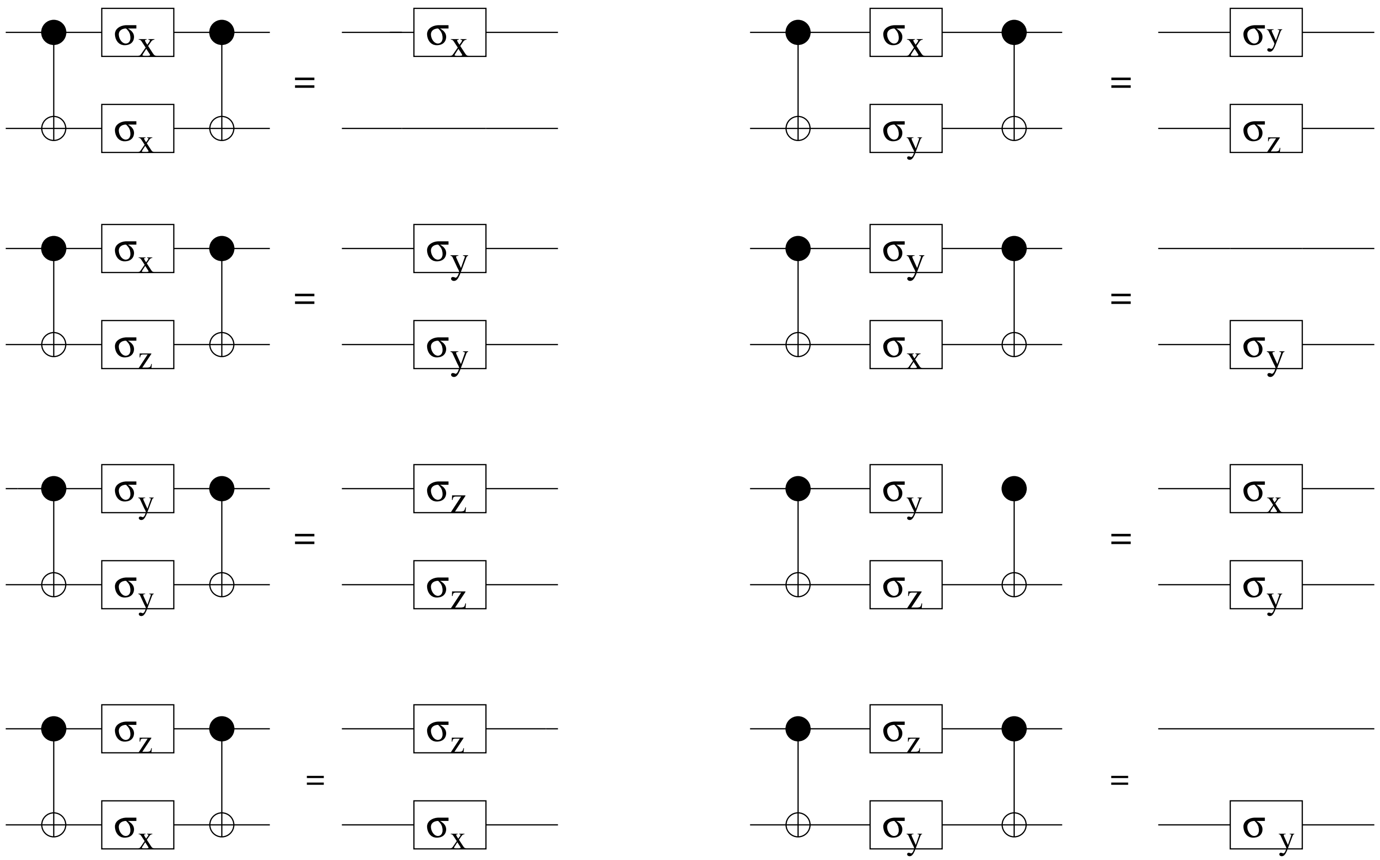


SOME USEFUL IDENTITIES

Several simple identities between elementary gates are surprisingly useful.



ANOTHER USEFUL IDENTITIES



PERMUTATION CIRCUITS

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.

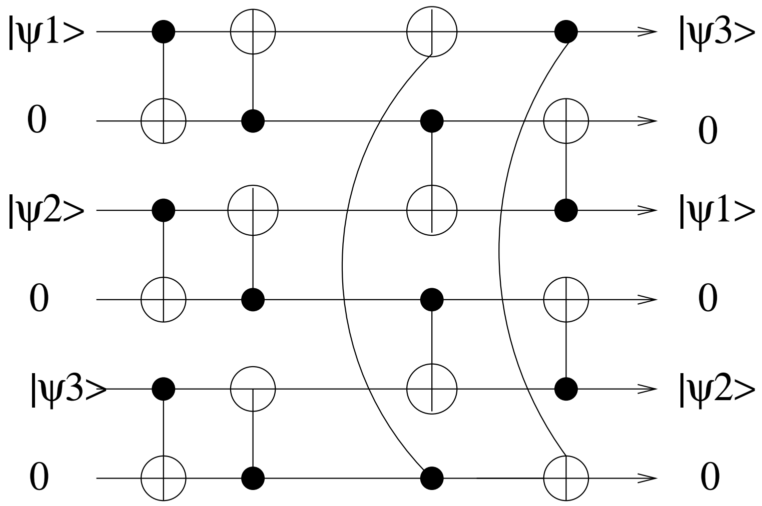
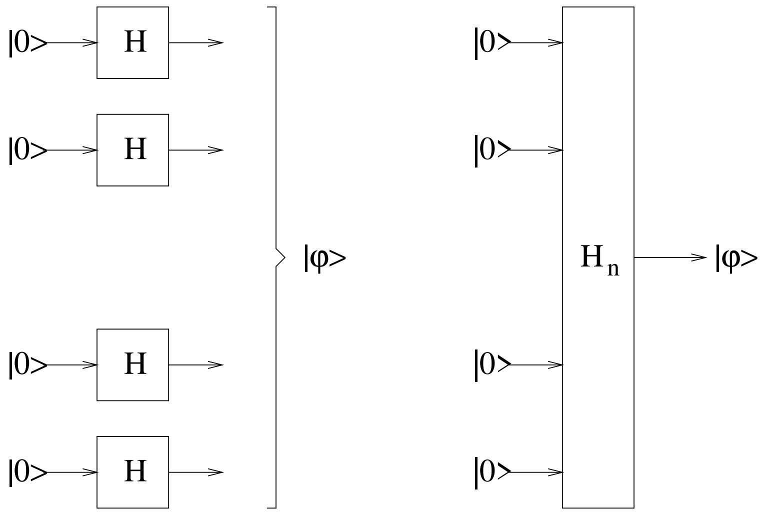


Figure 5: Permutation circuit

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.

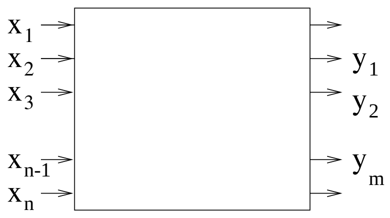


(a) (b)

The Hadamard circuit H_n and its application to the state $|0^{(n)}\rangle$ with the outcome

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

COMPUTATIONAL MEANING OF QGATES



The computational meaning of quantum circuits is defined as follows.

For any quantum circuit C with input variables $x_1, \ldots, x_n$ and output variables $y_1, \ldots, y_m$, $m \leq n$ (they are to be a subset of outputs), we associate to any input $x \in \{0,1\}^n$ the probability distribution ρ_x over $\{0,1\}^m$ defined in the following way.

For any input x the final quantum state v, corresponding to all output wires, not only to those carrying output variables, has the form

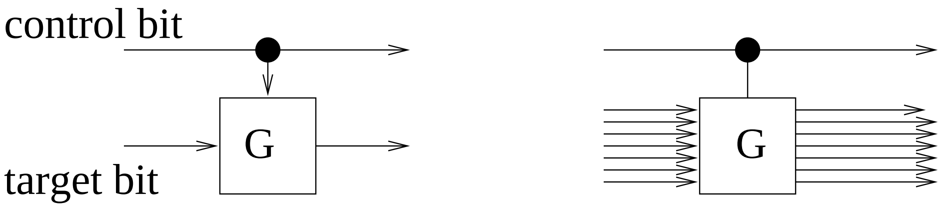
$$v = \sum_{y \in \{0,1\}^m} \alpha_y |y\rangle,$$

where α_y is the amplitude obtained by the projection of v when the output variables are set to the value y, i.e. α_y is the square root of the sum of squares of amplitudes of these final outcomes having value y in the wires corresponding to output variables.

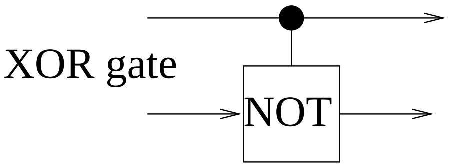
$\pi_x(y) = |\alpha_y|^2$ is the corresponding probability and $\{\pi_x \,|\, x \in \{0,1\}^n\}$ is said to be the distribution generated by the circuit C.

CONTROLLED QUANTUM GATES

To any quantum gate G we can design a **controlled version** CG of G as follows



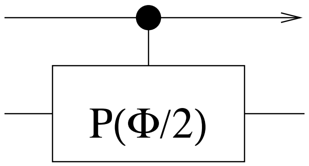
For example



If

$$R(\theta) = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \quad \text{or} \quad R(\theta) = \begin{pmatrix} \cos\theta & i\sin\theta \\ i\sin\theta & \cos\theta \end{pmatrix}$$

then



represents rotation of the target qubit by angle $\frac{\theta}{2}$ if the control bit is 1.

MEASUREMENT GATES

Measurement gates are not only to magnify results of quantum evolution to provide its outcomes to the classical world. They can be used also to influence, in an essential way, the whole process of quantum computation.

Consider the two quantum circuits depicted in Figure. The first one consists at first of two Hadamard gates H_n and ends with the measurement gate, with respect to the standard observable.

The second circuit has in addition a measurement gate also in between two Hadamard gates.

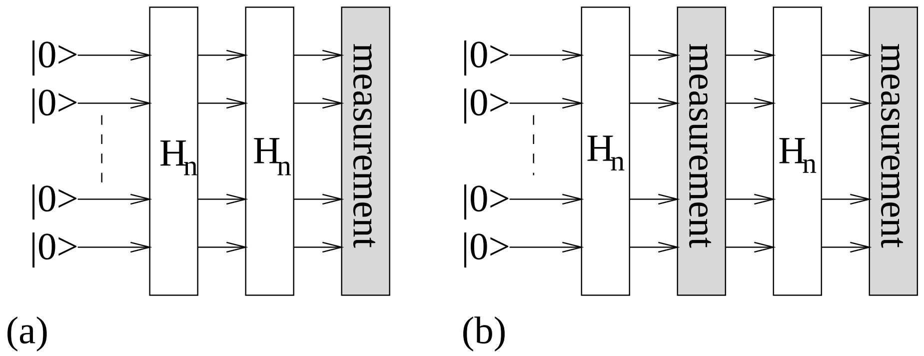
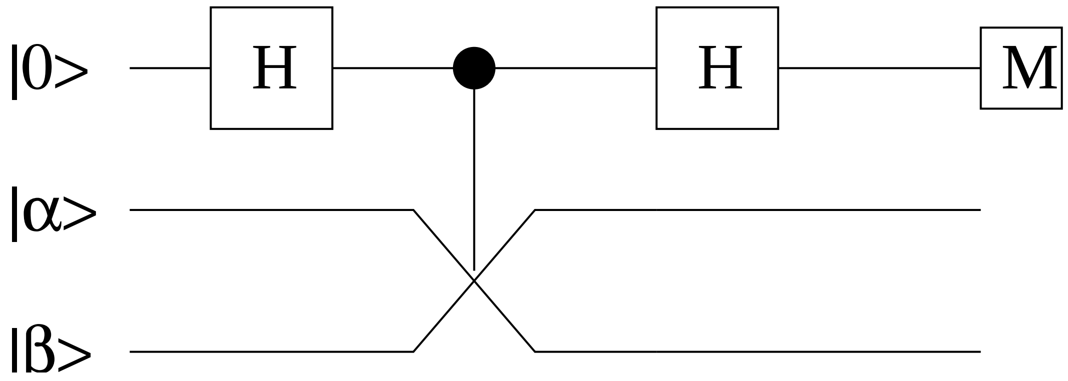


Figure 6: Measurement gates and their role

There is an essential difference between these two circuits. Which one?

QUBIT TESTING CIRCUIT

Let $|\alpha\rangle$ and $|\beta\rangle$ be two qubits and $\delta = |\langle\alpha|\beta\rangle|$. Equality of two qubits can be tested by the following so called swap test circuit



consisting of two Hadamard gates, one Fredkin gate (that exchanges inputs on other two inputs if target bit is 1 and otherwise let all inputs to get through without any change.
Indeed, the resulting state before the measurement is the state

$$\frac{1}{\sqrt{2}}(|0\rangle(|\alpha\rangle|\beta\rangle + |\beta\rangle|\alpha\rangle) + |1\rangle(|\alpha\rangle|\beta\rangle - |\beta\rangle|\alpha\rangle))$$

and therefore if $|\alpha\rangle = |\beta\rangle$, then the result of measurement is 0 with probability 1 and if $|\alpha\rangle \neq |\beta\rangle$, then the result of measurement is 0 with probability $(1+\delta^2)/2$ and the result of measurement is 1 with probability $(1-\delta^2)/2$.

BASIC OBSERVATIONS – II

- Nature offers many ways – let us call them technologies – various quantum information processing primitives can be exhibited, realized and utilized.
- Since it appears to be very difficult to exploit potential of nature for QIP, it is of large importance to explore which quantum primitives form universal sets of primitives, and are (quite) easy to implement.
- Also from the point of view of understanding of the laws and limitations of QIP and also o f quantum mechanics itself, the problems of finding rudimentary and universal QIP primitives , as well as methods for their optimal use, are of large experimental and fundamental importance.
- Search for quantum computation universal primitives, and their optimal use, is actually one of the major tasks of the current QIP research (both theoretical and experimental) that starts to attack the task of building quantum processors seriously.

SETS of UNIVERSAL PRIMITIVES in CLASSICAL COMPUTING

- In *classical computing*, the most often used universal sets of gates are
  - AND-, OR- and NOT-gates,
  - AND- and NOT-gates,
  - NOR- (NAND-) gate.
- The optimization problem for classical circuits with such sets of gates has been solved qu ite satisfactorily.
- In case of *classical reversible computing*, universal are both the Toffoli gate

$$T(x,y,z) = (x, y, (x \wedge y) \oplus z)$$

and the Fredkin gate

$$F(x,y,z) = (x, \bar{x}y + xz, \bar{x}z + xy),$$

if constant inputs are allowed, as well as “wires” with the identity gates.

BASIC CONCEPTS – APPROXIMABILITY

Definition An operator

$$U : \mathcal{H}_{2^r} \rightarrow \mathcal{H}_{2^r}$$

is ε-approximated, for an $\varepsilon > 0$, by an operator

$$\bar{U} : \mathcal{H}_{2^n} \rightarrow \mathcal{H}_{2^n},$$

where $n \geq r$, using an ancilla state $|\alpha\rangle \in \mathcal{H}_{2^{n-r}}$, if for any state $|\phi\rangle \in \mathcal{H}_{2^r}$,

$$||\bar{U}(|\phi\rangle \otimes |\alpha\rangle) - U(|\phi\rangle) \otimes |\alpha\rangle|| \leq \varepsilon.$$

TYPES of UNIVERSALITIES

Definition A set of gates $\mathcal{G}$ is called fully universal (f-universal) if every gate can be realized, up to a global phase factor, by a $\mathcal{G}$-circuit.

Definition A set of gates $\mathcal{G}$ is called universal if there is an integer n_0 such that any n-qubit unitary gate with $n \geq n_0$, can be, for any $\varepsilon > 0$, ε-approximated by a $\mathcal{G}$-circuit.

Definition A set of gates $\mathcal{G}$ is called densely universal (d-universal) if there exists an integer n_0 such that for any $n \geq n_0$, the subgroup generated by $\mathcal{G}$ is dense in $SU(2^n)$.

Definition A set of real gates $\mathcal{G}$ is called computationally universal (c-universal) if there is an integer n_0 such that any n-qubit real unitary gate with $n \geq n_0$, can be, for any $\varepsilon > 0$, ε-approximated by a $\mathcal{G}$-circuit.

BASIC GATES

Gates that will play an important role in the following:

$$\sigma_x = X, \sigma_y = Y, \sigma_z = Z, K = \sigma_z^{\frac{1}{2}}, T = \sigma_z^{\frac{1}{4}}.$$

where σ_x, σ_y and σ_z are Pauli operators;

$$CNOT = \Lambda_1(\sigma_x),\ \ \text{TOFFOLI} = \text{TOF} = \Lambda_2(\sigma_x),$$

and

$$\text{HADAMARD} = H = \frac{1}{\sqrt{2}}(\sigma_x + \sigma_z),\ \ \text{SWAP} \quad \text{and } \sqrt{\text{SWAP}}.$$

BASIC ROTATION GATES

Rotations around axes:

$$R_x(\theta) = e^{-i\theta\sigma_x/2} = \cos\frac{\theta}{2}I - i\sin\frac{\theta}{2}\sigma_x = \begin{pmatrix} \cos\frac{\theta}{2} & -i\sin\frac{\theta}{2} \\ -i\sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{pmatrix}$$

$$R_y(\theta) = e^{-i\theta\sigma_y/2} = \cos\frac{\theta}{2}I - i\sin\frac{\theta}{2}\sigma_y = \begin{pmatrix} \cos\frac{\theta}{2} & -\sin\frac{\theta}{2} \\ \sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{pmatrix}$$

$$R_z(\theta) = e^{-i\theta\sigma_z/2} = \cos\frac{\theta}{2}I - i\sin\frac{\theta}{2}\sigma_z = \begin{pmatrix} e^{-i\frac{\theta}{2}} & 0 \\ 0 & e^{i\frac{\theta}{2}} \end{pmatrix}$$

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

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

UNIVERSAL GATES

Definition 0.1 *A set of quantum gates is* **universal** *if any unitary transformation* U *on any qubit register can be performed, with arbitrary precision* $\varepsilon > 0$*, by a quantum circuit* $C_{U,\varepsilon}$*, consisting of the gates from that set. (In other words, the unitary matrix defined by* $C_{U,\varepsilon}$ *is* ε*-close to* U*.)*

A quantum gate is **universal** *if by itself it forms a universal set when supported by constant inputs* $|0\rangle$ *and* $|1\rangle$

SETS OF UNIVERSAL QUANTUM PRIMITIVES

Several examples of universal sets of quantum computation primitives are known.

Deutsch gate

$$D(\theta) = \begin{pmatrix} \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \begin{matrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & i\cos\theta & \sin\theta \\ 0 & 0 & \sin\theta & i\cos\theta \end{matrix} \end{pmatrix}.$$

Barenco gate

$$A(\phi, \alpha, \theta) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & e^{i\alpha}\cos\theta & -ie^{i(\alpha-\phi)}\sin\theta \\ 0 & 0 & -ie^{i(\alpha+\phi)}\sin\theta & e^{i\alpha}\cos\theta \end{pmatrix}.$$

A SIMPLE UNIVERSAL GATE

It is well known that any rotation on Bloch sphere can be composed out of rotations

$$R_y(\phi) = \begin{pmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{pmatrix} \qquad R_z(\phi) = \begin{pmatrix} e^{-i\phi} & 0 \\ 0 & e^{i\phi} \end{pmatrix}$$

and these gates can be also used to construct a universal 2-qubit gate (Tamir, 2004)

$$\begin{pmatrix} R_y(\alpha) & 0 \\ 0 & R_z(\beta) \end{pmatrix},$$

where α, β and π are linearly independent over rationals.

FUNDAMENTAL RESULTS

The first really satisfactory results, concerning universality of gates, have been due to Barenco et al. (1995)

Theorem 0.2 *CNOT gate and all one-qubit gates form a universal set of gates.*

The proof is in principle a simple modification of the RQ-decomposition from linear algebra. Theorem 0.2 can be easily improved:

Theorem 0.3 *CNOT gate and* elementary rotation gates

$$R_\alpha(\theta) = \cos\frac{\theta}{2}I - i\sin\frac{\theta}{2}\sigma_\alpha, \quad \textit{for} \ \ \alpha \in \{x, y, z\}.$$

form a universal set of gates.

An important generalization has been due to Brylinskis (2001)

Theorem 0.4 *Any entangling two-qubit gate and all one-qubit gates (or all elementary gates) form a universal set of gates.*

SETS of UNIVERSAL COMPUTATION PRIMITIVES

1. XOR gate and one-qubit gates;
2. A simple universal set of quantum gates consists of gates:

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

ENTANGLING GATES

A two-qubit gate is called **entangling** if it can create an entangled state when applied to a separable state.

Entangling gates are important. As shown by J. Brylinski and R. Brylinski (2001):

A two qubit gate forms with one-qubit gates a universal set of gates if and only if it is an entangling gate.

MAJOR FINITE UNIVERSAL SETS OF GATES

The following are finite, interesting and important d-universal sets of gates:

- SHOR$=\{$TOF$, H, \sigma_z^{\frac{1}{2}}\}$, see Shor (1996).
- KLZ1 $= \{CNOT, \Lambda_1(\sigma_z^{\frac{1}{2}}), \sigma_z^{\frac{1}{2}}\}$, see Knill et al. (1998?).
- KITAEV $= \{\Lambda_1(\sigma_z^{\frac{1}{2}}), H\}$, see Kitaev (1997).
- BMPRV$=\{CNOT, H, \sigma_z^{\frac{1}{4}}\}$, see Boykin et al. (1999).

Kitaev (1997) has shown universality of the set KITAEV. Since sets KITAEV and SHOR are equivalent and gates in SHOR can be simulated by KLZ1-circuits. Universality of the set KLZ1 follows from that.

COMPUTATIONALLY UNIVERSAL SETS OF GATES

- Bernstein and Vazirani (1993) have shown that for having universal quantum computation it is sufficient to work with real amplitudes.
- Adleman et al. (1997) have shown that the set of amplitudes that is really needed is very small, for example

$$A = \{0, \pm 3/5, \pm 4/5, \pm 1\}, \text{ or } B = \{0, \pm 1/\sqrt{2}, \pm 1\},$$

  or $C = \{0, \pm\cos\theta, \pm\sin\theta, \pm 1\}$, for various θ.

- Rudolph and Grover (2002) have shown, surprisingly, that a simple two-qubit real gate

$$G = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \cos\phi & -\sin\phi \\ 0 & 0 & \sin\phi & \cos\phi \end{pmatrix},$$

  with ϕ being an irrational multiple of π, is computationally universal.

SHI's RESULTS

Surprising results have been obtained by Shi (2003)

Theorem 0.5 • *Toffoli gate and any one-qubit gate changing the computational basis form a computationally universal set of gates.*

- *CNOT gate and any one-qubit gate such that its square does not preserve computational basis form a universal set of g ates.*

As a consequence

- Toffoli and Hadamard gates form a computationally universal set of gates.

Since Toffoli gate is universal for classical reversible computing, Shi's result means that full power of quantum computation is obtained by adding just the Hadamard gate.

BINARY ADDER

The following quantum circuit performs binary addition.

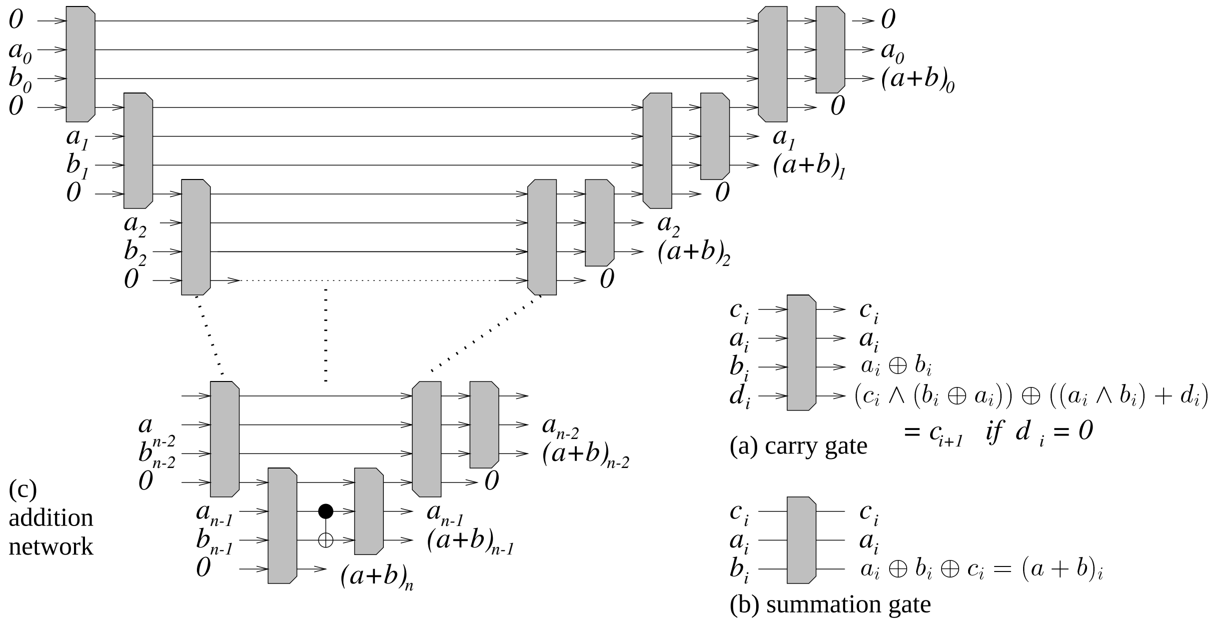


Figure 7: Quantum network for binary addition

PART II - OPTIMIZATION

EFFICIENCY of UNIVERSAL SETS of QUANTUM PRIMITIVES

- It is a natural and important question to ask how good are, from the efficiency point of view, different universal sets of quantum primitives.
- So called Solovay-Kitaev theorem implies that for evolutionary and computational universality, it is not costly to replace one universal basis by another one – it requires only poly-logarithmic overhead in $\lg 1/\varepsilon$ and that that number of base gates are needed.
- Solovay-Kitaev result implies that any gate from one finite universal set can be approximated with precision ε using polylog($\frac{1}{\varepsilon}$) gates from other finite universal set of gates. More exactly, Solovay and Kitaev showed that there exist polynomial time algorithm (in $\lg 1/\varepsilon$ that creates a circuit with $\mathcal{O}(\lg^c(1/\varepsilon))$ gates, where $c \in [3, 4]$.)

DECOMPOSITION of UNITARIES into ONE- and TWO-QUBIT GATES

Two very basic questions concerning decomposition of n-qubit unitaries into one-and two-qubit gates are the following ones

- **What is the total number of one- and two-qubits gates needed to decompose an arbitrary n qubit unitary operation?**
- **What is the total number of CNOT gates (or of some other entangling two qubit gates) needed to decompose an arbitrary n qubit unitary?**

GENERAL RESULTS

- **Barenco et al. (1995) have shown that any n qubit gate can be realized by $\mathcal{O}(n^3 4^n)$ CNOT and one-qubit gates.**
- **The above result has been improved, step by step, to $\mathcal{O}(n^2 4^n)$, $\mathcal{O}(n 4^n)$ and, finally, by Vartiainen et al. (2003) to $\mathcal{O}(4^n)$ – asymptotically tight.**
- **Concerning the CNOT gates only:**
  - **The best known upper bound is $\mathcal{O}(4^n)$ due to Vartiainen et al. (2003).**
  - **The best lower bound, due to Shende et al. (2003), is $\lceil (4^n - 3n - 1)/4 \rceil$.**

KEY PROBLEMS

- **The key problem is how many CNOT and one-qubit gates are necessary and sufficient to implement any two-qubit gate.**
- **Since each one-qubit gate can be expressed as a composition of any two of the elementary rotation gates R_x, R_y and R_z, it is of interest, and actually of large practical importance, to determine what is the minimal number of (elementary) gates R_x, R_y, R_z and $CNOT$ needed to implement an arbitrary two-qubit gate.**

MAIN OUTCOMES

We discuss here only the best outcomes, so far, mainly due to Vidal and Dawson (2003), Shende et al. (2003) and Vatan and Williams (2003).

- 3 CNOT gates and 10 one-qubit and CNOT gates in total are sufficient to realize any two qubit gate.
- 3 CNOT gates and 9 gates in total are necessary.
- Each two-qubit gate can be realized using 3 CNOT gates and in total with 18 gates from the set containing the CNOT gate and any two of the three gates from the set $\{R_x, R_y, R_z\}$. (The above result is optimal for the case temporary storage is not allowed (because of being expensive).
- For gates from $SO(4)$ only 12 gates R_y, R_z are needed.

UNIVERSAL CIRCUIT SCHEMES

The universal two-qubit circuit scheme with three CNOT gates and 10 basic gates, or 18 gates from the set $\{CNOT, R_y, R_z\}$ is in Fig ure 8.

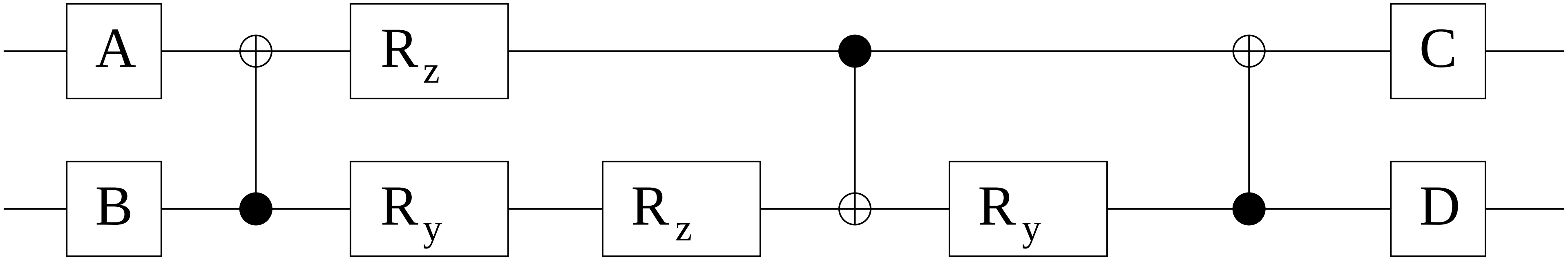


Figure 8: A universal 2-qubit circuit

B-GATE STORY

Search for the best implementation of two qubit gates using a fixed two-qubit gate and one-qubit gates brought also a discovery of a new gate, so called **B-gate**. It is the gate realized by the following circuit:

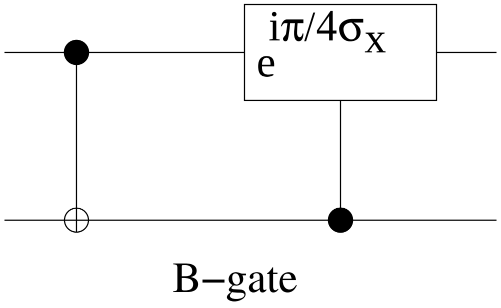


Figure 9: B-gate circuit

This gate is "better" than CNOT gate in the following sense.

Theorem Each two-qubit gate can be realized by a circuit with at most two B-gates and one-qubit gates.

MODELS of UNIVERSAL COMPUTERS

- Classical models: circuits, Turing machines, cellular automata, RAM a PRAM
- Quantum models
  - (Unitary operations based) Quantum Turing Machines
  - (Unitary operations based) Quantum Circuits - all gates are unitaries
  - Quantum cellular automata ????
  - Measurements based quantum circuits - all gates are measurements;
  - Measurements based quantum Turing machines (Perdrix, Jorrand, 2004);

All these models have the same (Turing) computational power.

TWO COMPUTATION MODES

Initialize $\longrightarrow$ Compute $\longrightarrow$ Get a results

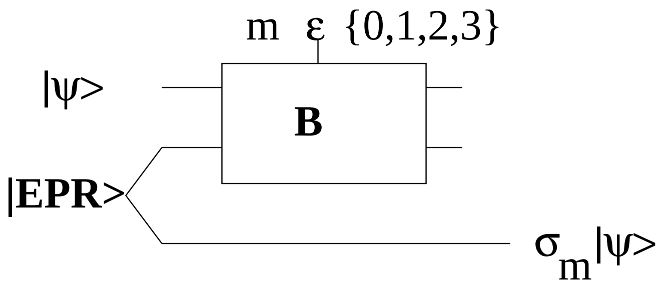
Initial state preparation $\longrightarrow$ $\boxed{\text{unitary operation}}$ $\longrightarrow$ measurement

measurements $\longrightarrow$ measurements $\longrightarrow$ measurements

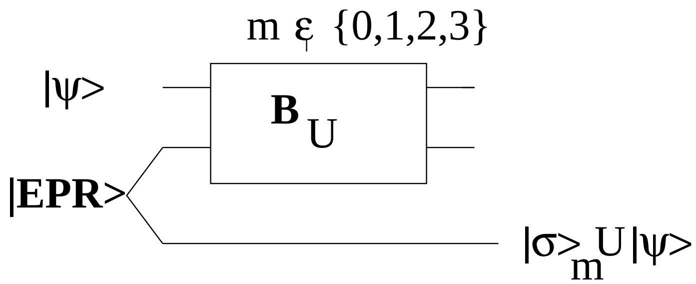
Measurement = projective measurement.

GENERALIZED TELEPORTATION

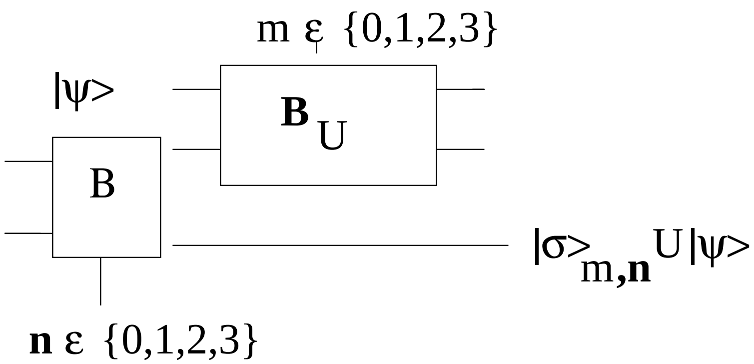
Teleportation



Teleportation of any unitary U



for a proper measurement B_U. In a slightly different form this looks as follows:



MINIMAL RESOURCES for UNIVERSAL MEASUREMENTS

Perdrix (2004) has shown that

one-qubit ancilla and

one two-qubit Pauli measurement ($X \otimes Z$) and three one-qubit Pauli measurements (X, Z, $\frac{1}{\sqrt{2}}(X + Y)$)

are sufficient to approximate, up to a Pauli operator, any unitary operation.

MINIMAL RESOURCES for UNIVERSAL MEASUREMENTS

Perdrix (2004) has shown that one-qubit ancilla, one two-qubit Pauli measurement and three one-qubit Pauli measurements are sufficient to approximate, up to a Pauli operator, any unitary operation.

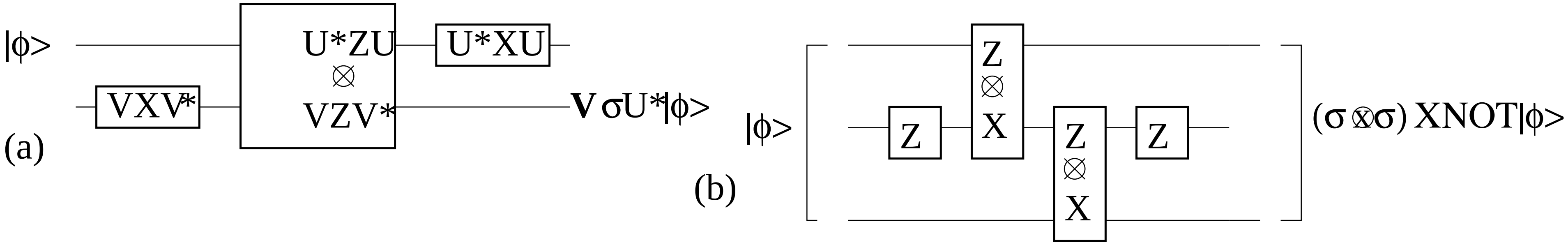


Figure 10: Two schemes for providing universal state transfer

- In the cases $U = H$ and $V = I$, the output has the form $\sigma H|\phi\rangle$.
- In the case $U = T = \sigma_z^{\frac{1}{4}}$ and $V = H$ the output has the form $\sigma HT|\phi\rangle$.
- In the above two cases only the measurements with observables X, Z, $\frac{1}{\sqrt{2}}(X+Y)$ and $X \otimes Z$ are used.