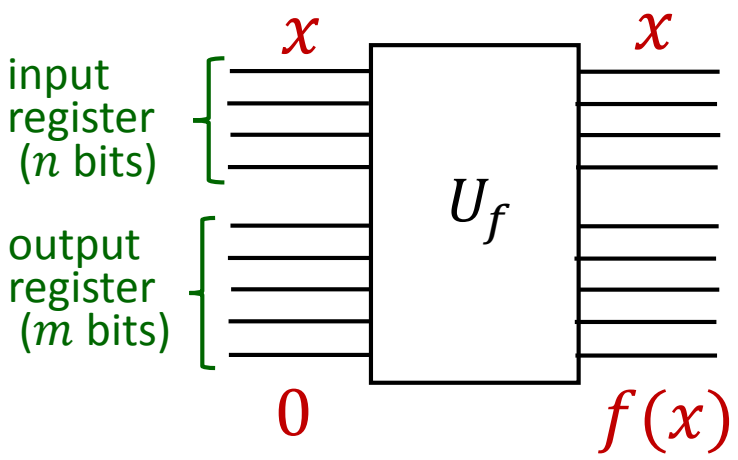
Usual Quant.Comp. protocol for a function calculation

Classically: $x \longrightarrow f(x)$

n bits $\quad$ m bits

Need reversible classical computation (since unitary); idea: store input

reversible implementation



In QC

$$U_f|x\rangle_n|y\rangle_m = |x\rangle_n|y \oplus f(x)\rangle_m$$

here # of qubits

bitwise addition modulo 2 (bitwise XOR)

In particular,

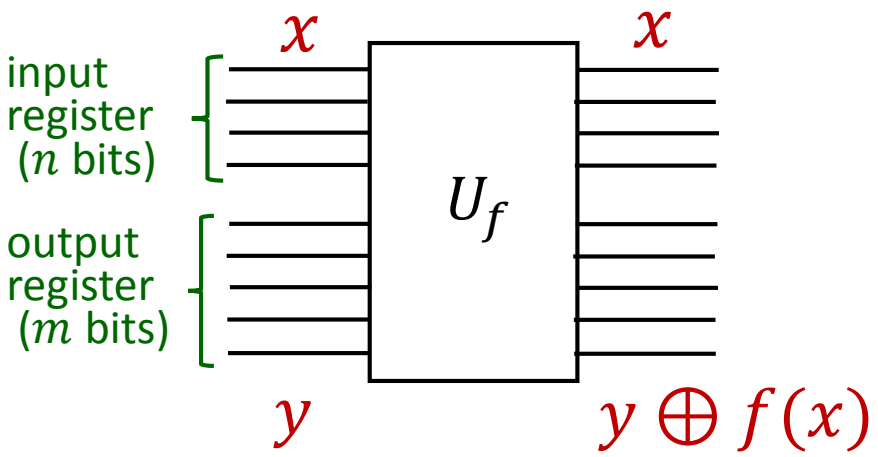
$$U_f|x\rangle_n|0\rangle_m = |x\rangle_n|f(x)\rangle_m$$

Obviously reversible: $U_f^2|x\rangle_n|y\rangle_m = |x\rangle_n|y \oplus f(x) \oplus f(x)\rangle = |x\rangle_n|y\rangle_m$

$$U_f^2 = \hat{1} \quad \Rightarrow \quad U_f^{-1} = U_f$$

<u>Permutation</u> of basis vectors in computational basis $\Rightarrow$ obviously unitary

Usual QC protocol for a function calculation (cont.)



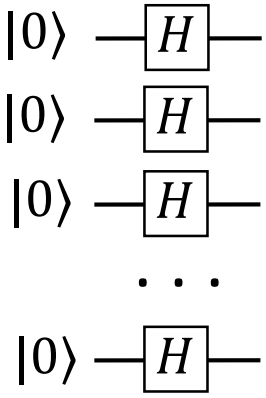
How to construct?

In principle, always possible: classical → classical reversible → quantum gates instead of classical gates

Problem: garbage (nuisance in classical case, critical in quantum case); however, garbage can always be restored to $|0\rangle$ (will discuss later)

Of course, repeating classical circuit design is not most efficient $\Rightarrow$ art of design

Main trick in function calculation



two qubits

$$(H \otimes H)\,|0\rangle|0\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}}\,\frac{|0\rangle + |1\rangle}{\sqrt{2}} = \frac{|00\rangle + |01\rangle + |10\rangle + |11\rangle}{\sqrt{4}} = \frac{1}{\sqrt{4}}\sum_{x=0}^{3}|x\rangle$$

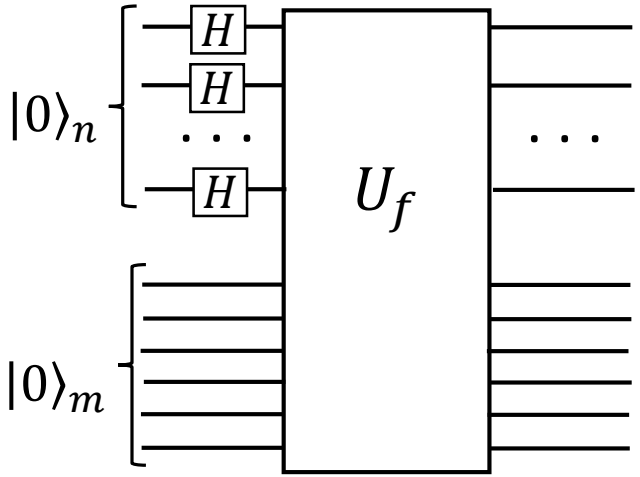
n qubits

$$H^{\otimes n}|0\rangle_n = \frac{1}{\sqrt{2^n}}\sum_{x=0}^{2^n-1}|x\rangle_n$$

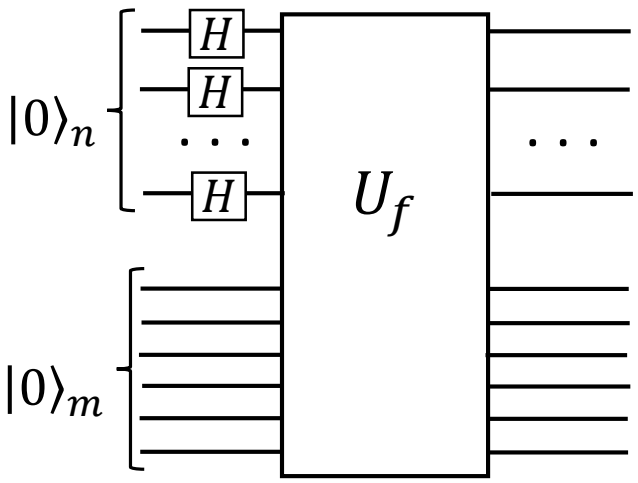
$\underbrace{H \otimes H \otimes \ldots H}_{n}$ $\underbrace{|0\rangle|0\rangle \ldots |0\rangle}_{n}$

contains all numbers in equal superposition

Therefore use as the input
(all inputs at the same time!)



Main trick in function calculation (cont.)



Formally,

$$U_f\left(H^{\otimes n} \otimes \hat{1}_m\right)\left(|0\rangle_n|0\rangle_m\right) =$$

$$= \frac{1}{2^{n/2}} \sum_{x=0}^{2^n-1} U_f\left(|x\rangle_n|0\rangle_m\right) =$$

$$= \frac{1}{2^{n/2}} \sum_{x=0}^{2^n-1} |x\rangle_n|f(x)\rangle_m$$

contains all functions at once!

Called quantum parallelism (for 1,000 qubits very impressive)

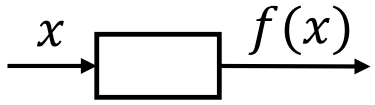
But if we measure all qubits, then get a random x and corresponding $f(x)$.
The same as in classical computer for random $x \Rightarrow$ not useful.
Also, cannot copy result (no cloning).

However, some further operations can give some relations between values of function $f(x)$ for different x. $\Rightarrow$ Art of quantum algorithms.

Deutsch's problem (David Deutsch, 1985)

A very simple toy problem

Black box, 1 bit → 1 bit

Classical 

4 possible functions

	$f(0)$	$f(1)$
f_0	0	0
f_1	0	1
f_2	1	0
f_3	1	1

If input 0, then can distinguish (f_0, f_1) from (f_2, f_3)

If input 1, then can distinguish (f_0, f_2) from (f_1, f_3)

However, we are interested in the question: constant or balanced?

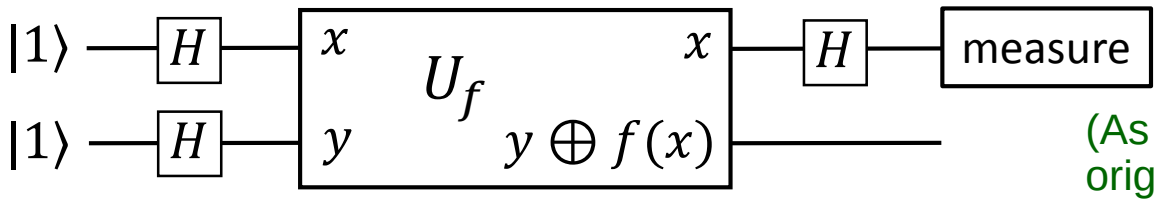
f_0, f_3 f_1, f_2

Classically, cannot find out in one query, necessarily need two queries.

With QC, only one query needed.

(Not quite interesting, but also not completely uninteresting if $f(x)$ is hard to calculate, for example, millionth bit in binary expression of $\sqrt{2+x}$ -- Mermin's book).

Deutsch's problem (cont.)



(As in Mermin's book; differs from original Deutsch's solution, in which success was only in half of cases)

Slightly differs from our usual way for input (starts with $|1\rangle$ instead of $|0\rangle$)

$$|11\rangle \xrightarrow{HH} \frac{|0\rangle - |1\rangle}{\sqrt{2}} \frac{|0\rangle - |1\rangle}{\sqrt{2}} = \frac{|0\rangle|0\rangle - |0\rangle|1\rangle - |1\rangle|0\rangle + |1\rangle|1\rangle}{2} \rightarrow$$

$$\xrightarrow{U_f} \frac{|0\rangle\,|f(0)\rangle - |0\rangle\left|\overline{f(0)}\right\rangle - |1\rangle|f(1)\rangle + |1\rangle\left|\overline{f(1)}\right\rangle}{2} =$$

$\bar{f} = 1 - f$
$= 1 \oplus f$

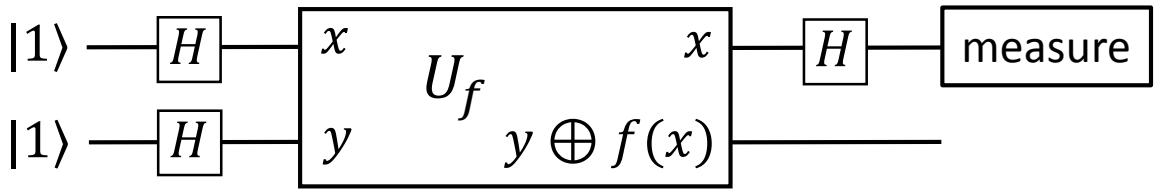
$$= \begin{cases} \text{constant,} & \dfrac{(|0\rangle - |1\rangle)\,|f(0)\rangle - (|0\rangle - |1\rangle)\left|\overline{f(0)}\right\rangle}{2} = \dfrac{(|0\rangle - |1\rangle)\,(\,|f(0)\rangle - \left|\overline{f(0)}\right\rangle)}{2} \rightarrow \\ \text{balanced,} & \dfrac{(|0\rangle + |1\rangle)\,|f(0)\rangle - (|0\rangle + |1\rangle)\left|\overline{f(0)}\right\rangle}{2} = \dfrac{(|0\rangle + |1\rangle)\,(\,|f(0)\rangle - \left|\overline{f(0)}\right\rangle)}{2} \rightarrow \end{cases}$$

$$\xrightarrow{H} \begin{cases} \text{constant,} & |1\rangle \dfrac{|f(0)\rangle - \left|\overline{f(0)}\right\rangle}{\sqrt{2}} \\ \text{balanced,} & |0\rangle \dfrac{|f(0)\rangle - \left|\overline{f(0)}\right\rangle}{\sqrt{2}} \end{cases}$$

Therefore, measuring the first qubit, we find if the function was constant or balanced

Second qubit does not contain <u>any</u> information about f. (Seems to depend on $f(0)$, but actually only an overall phase.)

Deutsch's problem (cont.)



Another way to analyze

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

U_f either does nothing or exchanges $|0\rangle \leftrightarrow |1\rangle$, which flips sign of $(|0\rangle-|1\rangle)/\sqrt{2}$.

as if U_f acts on the first qubit

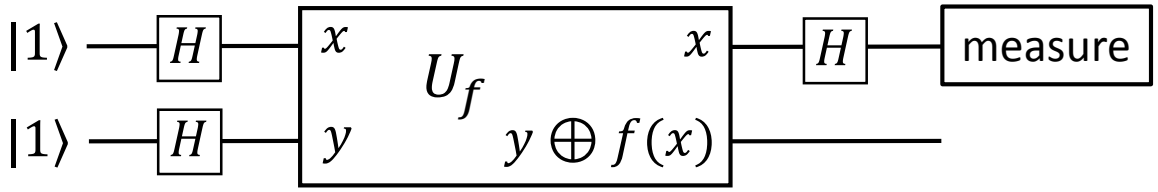
So consider only first (upper) qubit

$$|1\rangle \xrightarrow{H} \frac{|0\rangle-|1\rangle}{\sqrt{2}} \xrightarrow{U_f} \frac{(-1)^{f(0)}|0\rangle-(-1)^{f(1)}|1\rangle}{\sqrt{2}}$$

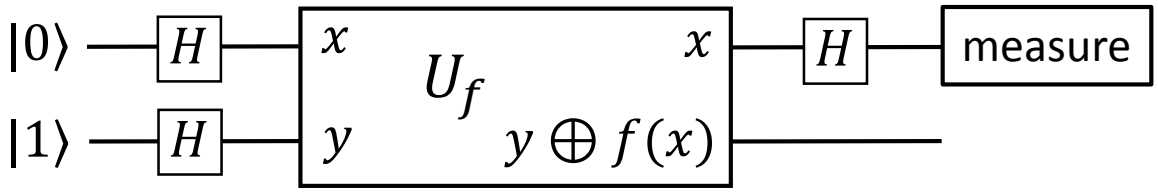
If $f(0)=f(1)$, then the same sign, $\pm\frac{|0\rangle-|1\rangle}{\sqrt{2}} \xrightarrow{H} \pm|1\rangle$

If $f(0)\neq f(1)$, then opposite sign, $\pm\frac{|0\rangle+|1\rangle}{\sqrt{2}} \xrightarrow{H} \pm|0\rangle$

Deutsch's problem (cont.)



Slight modification possible



The same, but start with $|0\rangle|1\rangle$

(in Nielsen-Chuang)

Then the first qubit evolves as

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

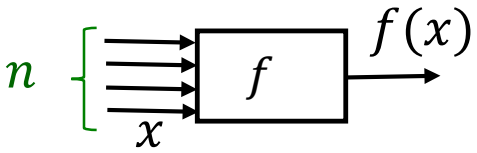
$$= \begin{cases} \text{constant,} & \pm\dfrac{|0\rangle + |1\rangle}{\sqrt{2}} \xrightarrow{H} \pm|0\rangle \\ \text{balanced,} & \pm\dfrac{|0\rangle - |1\rangle}{\sqrt{2}} \xrightarrow{H} \pm|1\rangle \end{cases}$$

very similar

Deutsch-Jozsa algorithm (1992, Cleve et al., 1998)

Somewhat similar to Deutsch problem, but now function f: n bit $\rightarrow$ 1 bit

$$f\colon \ \{0,1\}^n \rightarrow \{0,1\}$$



It is known that the function f is either constant or balanced

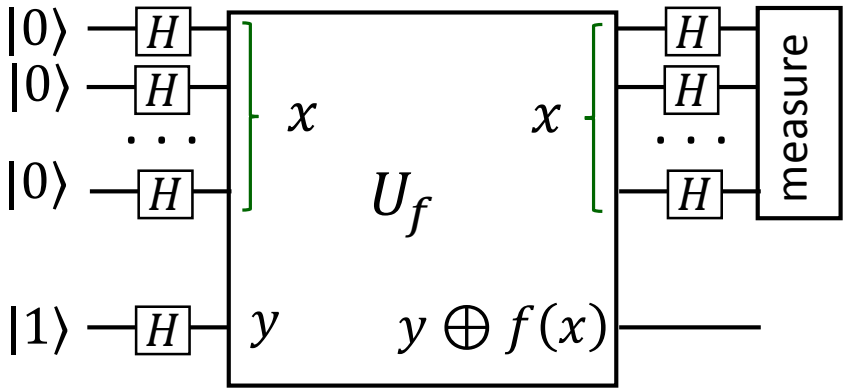
always 0 or always 1

# of 0s in the table is the same as # of 1s

Goal: constant or balanced?

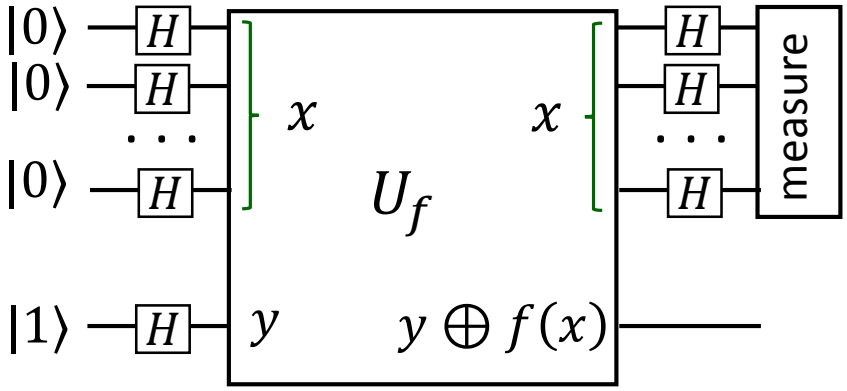
Classically, up to $2^{n-1}+1$ queries needed (especially if cheating, changing the black box)

With a quantum computer, just one query (exponential speed-up).



If result is $00..0$,
then f is constant,
if anything else,
then f is balanced

Deutsch-Jozsa algorithm (cont.)



If result is $00..0$, then f is constant,
if anything else, then f is balanced

$$|0\rangle^{\otimes n}|1\rangle \xrightarrow{H^{\otimes n}H} \frac{\sum_{x=0}^{2^n-1}|x\rangle}{\sqrt{2^n}} \frac{|0\rangle-|1\rangle}{\sqrt{2}} \xrightarrow{U_f} \frac{\sum_x(-1)^{f(x)}|x\rangle}{\sqrt{2^n}} \frac{|0\rangle-|1\rangle}{\sqrt{2}}$$

The same trick: change of the last qubit can be translated into change of the input register state

If function $f(x)$ is constant, then overall phase "+" or "−", but nothing changed
$\Rightarrow$ $H^{\otimes n}$ reverses evolution $\Rightarrow |0\rangle^{\otimes n}$.

If function $f(x)$ is balanced, then 2^{n-1} pluses and 2^{n-1} minuses in the sum.
Then after $H^{\otimes n}$ there is no contribution from $|0\rangle^{\otimes n}$, as will be shown next.

Deutsch-Jozsa algorithm (cont.)

Lemma

$$H^{\otimes n}|x_{n-1}..x_1x_0\rangle = \frac{\sum_{z_0,z_1,..z_{n-1}}(-1)^{x_0z_0+x_1z_1+\ldots+x_{n-1}z_{n-1}}\,|z_{n-1}\ldots z_1z_0\rangle}{\sqrt{2^n}}$$

Shorter notation:

$$\boxed{H^{\otimes n}|x\rangle = \frac{\sum_z(-1)^{x\cdot z}\,|z\rangle}{\sqrt{2^n}}}$$

$x \cdot z \equiv x_0z_0 \oplus x_1z_1 \oplus \ldots \oplus x_{n-1}z_{n-1}$ bitwise inner product modulo 2
(kind of an inner product, not really)

Special case For $|x\rangle = |0\rangle_n$ this reduces to $H^{\otimes n}|0\rangle = (\sum_z|z\rangle)/\sqrt{2^n}$ (already know).

Proof For one qubit $H|0\rangle = (|0\rangle + |1\rangle)/\sqrt{2}$, $H|1\rangle = (|0\rangle - |1\rangle)/\sqrt{2}$, which is

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

For n qubits

$$H^{\otimes n}|x_{n-1}\ldots x_1x_0\rangle = \frac{\sum_{z_{n-1}=0}^{1}(-1)^{x_{n-1}\cdot z_{n-1}}\,|z_{n-1}\rangle}{\sqrt{2}}\ \ldots\ \frac{\sum_{z_0=0}^{1}(-1)^{x_0\cdot z_0}\,|z_0\rangle}{\sqrt{2}} =$$

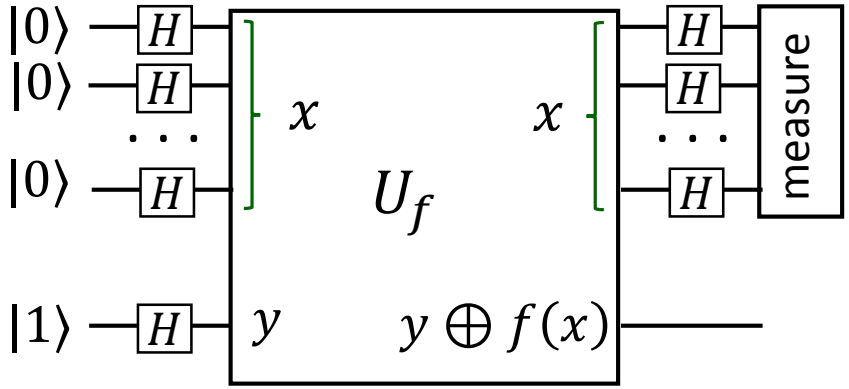
$$= \frac{1}{\sqrt{2^n}}\ \ldots\ \sum_{z_0,\ldots z_{n-1}}(-1)^{x_0z_0+\ldots+x_{n-1}z_{n-1}}\,|z_n\rangle\ldots|z_0\rangle$$

QED

Deutsch-Jozsa algorithm (cont.)

Now back to Deutsch-Jozsa

$$H^{\otimes n}|x\rangle = \frac{\sum_z (-1)^{x\cdot z}|z\rangle}{\sqrt{2^n}}$$



$$|0\rangle^{\otimes n}|1\rangle \xrightarrow{H^{\otimes n}H} \frac{\sum_{x=0}^{2^n-1}|x\rangle}{\sqrt{2^n}} \frac{|0\rangle-|1\rangle}{\sqrt{2}} \xrightarrow{U_f} \frac{\sum_x (-1)^{f(x)}|x\rangle}{\sqrt{2^n}} \frac{|0\rangle-|1\rangle}{\sqrt{2}}$$

$$\frac{\sum_x (-1)^{f(x)}|x\rangle}{\sqrt{2^n}} \xrightarrow{H^{\otimes n}} \frac{\sum_x \sum_z (-1)^{f(x)}(-1)^{x\cdot z}|z\rangle}{2^n}$$

Therefore, the amplitude for $|z\rangle = |0\rangle^{\otimes n}$ is $\frac{\sum_x (-1)^{f(x)}}{2^n} = \begin{cases} 0, & \text{balanced} \\ \pm 1, & \text{constant} \end{cases}$

So, if result is $00..0$, then $f(x)$ is constant,
if anything else, then $f(x)$ is balanced

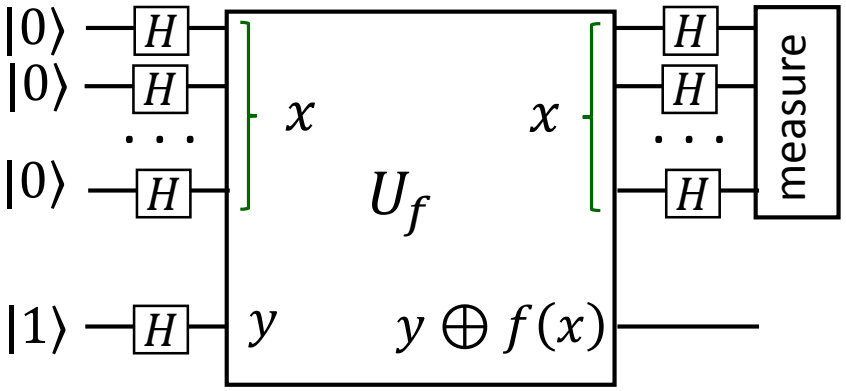
Bernstein-Vazirani algorithm (1993, 1997)

Black box, which calculates $f(x) = a \cdot x = a_0 x_0 \oplus a_1 x_1 \oplus \ldots \oplus a_{n-1} x_{n-1}$ for an unknown n-bit number a. Goal: to find a. How many queries needed?

$$f: \ \{0,1\}^n \to \{0,1\}$$

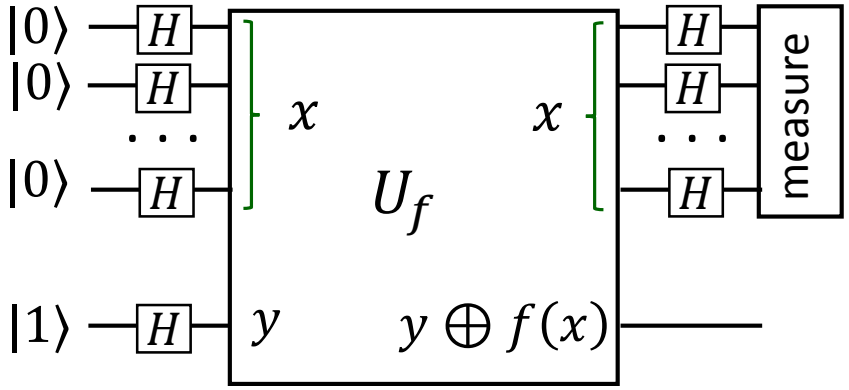
Classically: *n* queries (e.g., choose $x = 2^m$, $m = 0, 1, .. n-1$)

QC: just once, use the same circuit as for Deutsch-Jozsa



Measurement gives a directly

Bernstein-Vazirani algorithm (cont.)



$f(x) = a \cdot x$

Goal: to find a.

$$\left(H^{\otimes n} \otimes \hat{1}\right) U_f \left(H^{\otimes n} \otimes H\right) \left(|0\rangle_n |1\rangle\right)$$

$$|0\rangle^{\otimes n}|1\rangle \xrightarrow{H^{\otimes n}H} \frac{\sum_{x=0}^{2^n-1}|x\rangle}{\sqrt{2^n}} \frac{|0\rangle - |1\rangle}{\sqrt{2}} \xrightarrow{U_f} \frac{\sum_x (-1)^{f(x)}|x\rangle}{\sqrt{2^n}} \frac{|0\rangle - |1\rangle}{\sqrt{2}} \rightarrow$$

$$\xrightarrow{H^{\otimes n}} \frac{\sum_x \sum_z (-1)^{f(x)}(-1)^{x\cdot z}|z\rangle}{2^n} \frac{|0\rangle - |1\rangle}{\sqrt{2}} = |a\rangle_n \frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

$\Rightarrow$ get a when measure input register

Coefficient for $|z\rangle$: $\sum_x (-1)^{a\cdot x}(-1)^{z\cdot x} = \sum_{x_0,x_1,\ldots} \prod_j (-1)^{(a_j+z_j)x_j} =$

factorizes

$$= \prod_j \underbrace{\sum_{x_j=0}^{1} (-1)^{(a_j+z_j)x_j}}$$

0 if $a_j \oplus z_j = 1$
2 if $a_j \oplus z_j = 0$

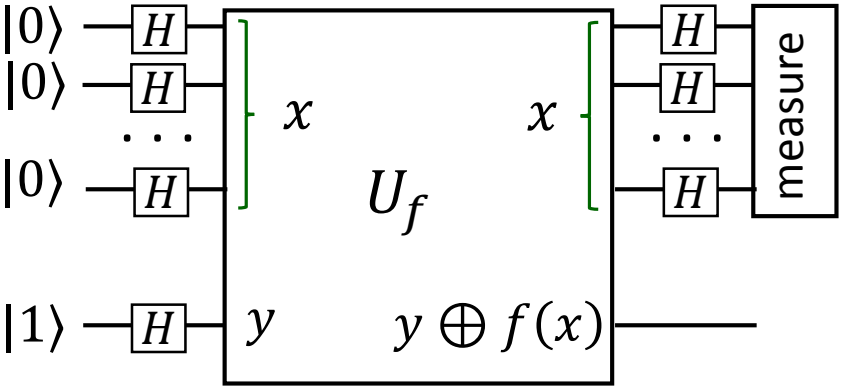
So, coefficient for $|z\rangle$ is non-zero only if $a_j \oplus z_j = 0$ for all j, i.e., if $z = a$. (Then coefficient 2^n, which cancels denominator.)

Bernstein-Vazirani algorithm (cont.)

Another way to analyze:

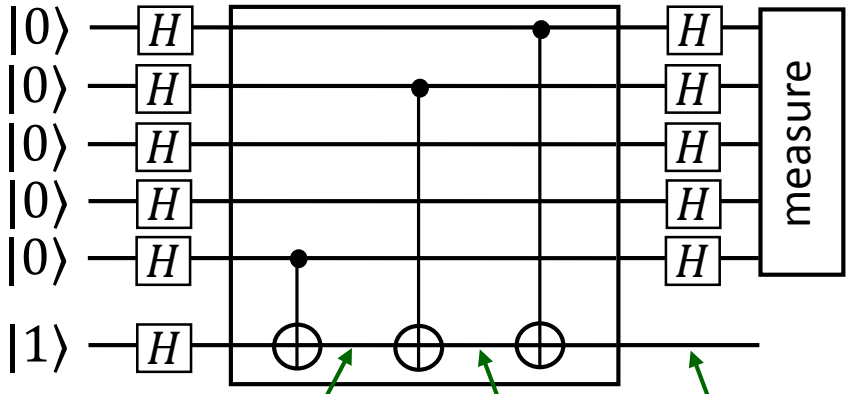
Use explicit realization of U_f with a quantum circuit diagram (any other realization is equivalent)

$$f(x) = a \cdot x$$



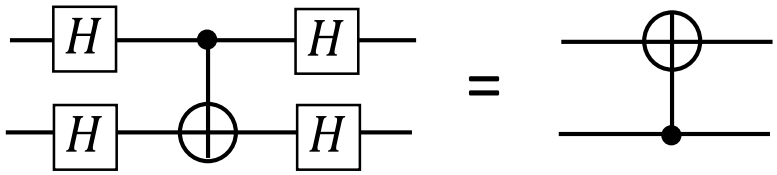
CNOT when $a_j = 1$

bottom qubit is flipped when $a_j x_j = 1$

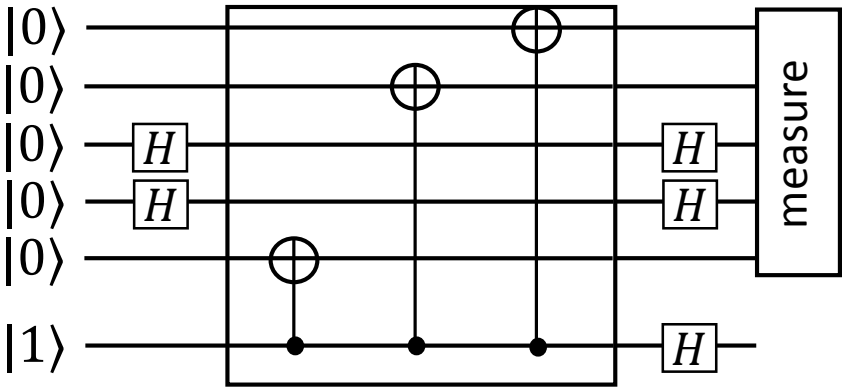


add

use



(proved earlier)



Now it is obvious that output is

$$|a\rangle_n \frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

Simon's problem (Daniel Simon, 1994)

Exponential speed-up (as in Deutsch-Jozsa), probabilistic, needs post-processing
(has some features of Shor's algorithm)

Function f: $\{0,1\}^n \to \{0,1\}^{n-1}$ (mapping exactly $2 \to 1$)

$f(x) = f(y)$ iff $x = y \oplus a$ for some (unknown) a $(a \neq 0)$

if and only if — bitwise XOR

Goal: find a

In some sense the goal is to find the period of f (as in Shor's algorithm)

Classically: try different x until the same output number, then we know a

If random choice of x, then in the worst case $2^{n-1} + 1$ queries.

Typically much less: after m trials the probability not to succeed is

$$P_{no\ success} \approx \left(1 - \frac{1}{2^n}\right)\left(1 - \frac{2}{2^n}\right) \cdots \left(1 - \frac{m-1}{2^n}\right) \approx \exp\left[-\frac{1}{2^n} - \frac{2}{2^n} - \cdots - \frac{m-1}{2^n}\right]$$

$$= \exp\left[-\frac{m\,(m-1)}{2 \times 2^n}\right]$$

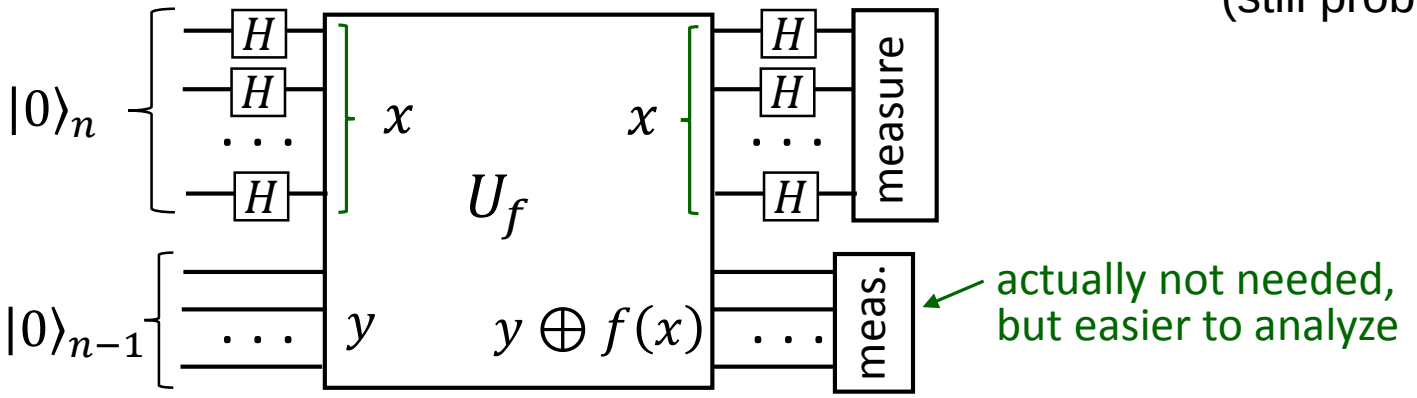
So, typical number of queries is $m \sim \sqrt{2^{n+1}} \sim 2^{n/2}$.
Much less than in the worst case, but still exponential in n.

Another derivation: $a \neq x_i \oplus x_j$, so m trials eliminate up to $m\,(m-1)/2$ values (pairs)

Some improvements possible (eliminate $x_i \oplus x_j \oplus x_k$), but still the same scaling

Simon's problem (cont.)

In contrast to $\sim 2^{n/2}$ queries classically, a QC can find a with $\sim n$ queries (still probabilistically)



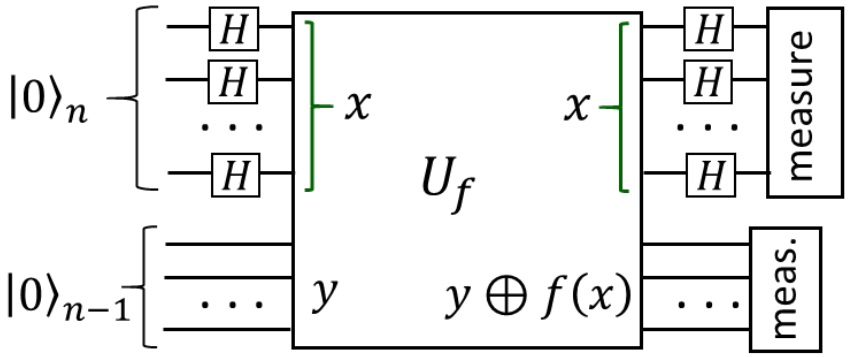
$$|0\rangle_n|0\rangle_{n-1} \xrightarrow{H^{\otimes n} \otimes \hat{1}_{n-1}} \frac{\sum_{x=0}^{2^n-1}|x\rangle}{\sqrt{2^n}}|0\rangle_{n-1} \xrightarrow{U_f} \sum_{x=0}^{2^n-1}|x\rangle_n|f(x)\rangle_{n-1}$$

If we measure output register, we get some value $|..\rangle_{n-1}$, and this collapses input register to contain only 2 values: x_0 and $x_0 \oplus a$, i.e., $(|x_0\rangle + |x_0 \oplus a\rangle)/\sqrt{2}$.
If cloning were possible, we would easily find a; however cloning is impossible.

$$\frac{|x_0\rangle + |x_0 \oplus a\rangle}{\sqrt{2}} \xrightarrow{H^{\otimes n}} \frac{1}{\sqrt{2}}\frac{1}{\sqrt{2^n}}\sum_{z=0}^{2^n-1}\left[(-1)^{x_0\cdot z} + (-1)^{(x_0\oplus a)\cdot z}\right]|z\rangle$$

$$= \frac{1}{\sqrt{2^{n+1}}}\sum_{z=0}^{2^n-1}(-1)^{x_0\cdot z}\underbrace{[1+(-1)^{a\cdot z}]}_{0 \text{ or } 2}|z\rangle = \frac{1}{\sqrt{2^{n-1}}}\sum_{z,\, a\cdot z=0}(-1)^{x_0\cdot z}|z\rangle$$

Simon's problem (cont.)



$$\frac{1}{\sqrt{2^{n-1}}} \sum_{z,\, a\cdot z=0} (-1)^{x_0 \cdot z} \, |z\rangle$$

$$|f(x_0)\rangle_{n-1}$$

So, measurement of input register will give some z, for which

$$a \cdot z = 0 = a_0 z_0 \oplus a_1 z_1 \oplus \ldots \oplus a_{n-1} z_{n-1}$$

This divides the space of possible a in half.

Do again, get another z, again divides the space of possible a in half. And so on.

$\Rightarrow$ In about n trials, we get unique a.

More exactly, after $n + \Delta$ runs, the probability of success is $p > 1 - 2^{-\Delta}$.

Calculation of a can be done by Gauss elimination in the system of linear equations: take one equation with coefficient 1 for a_0 and add to it (modulo 2) all other equations with coefficients 1 for a_0 (to get coefficient 0 for a_0); then the same for a_1, etc.

If output register is not measured, then $\frac{1}{\sqrt{2^{n-1}}} \frac{1}{\sqrt{2^{n-1}}} \sum_{x_0 < x_0 \oplus a} \sum_{z \cdot a = 0} (-1)^{x_0 \cdot z} \, |z\rangle |f(x_0)\rangle$

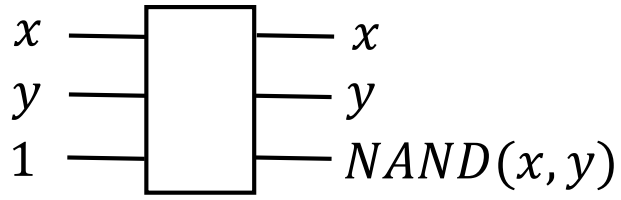
Same probabilities for all z, for which $a \cdot z = 0$, so meas. of output register is not needed.

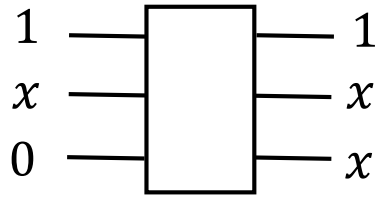
How to construct U_f for an arbitrary $f(x)$

In principle (not optimal) can always be done in the following way:

1) Classical $f(x)$ realized with AND, OR, NOT, NOR, NAND, XOR, etc.
2) Classical $f(x)$ realized in reversible way (using Toffoli)
3) Quantum with Toffoli
4) If desired, quantum Toffoli can be realized with CNOTs and 1-qubit gates
5) Garbage collection

Classical Toffoli: Controlled-Controlled-NOT (flips the third bit if both upper bits are 1)

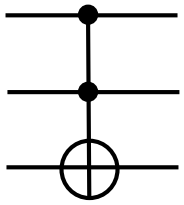




Fan-out

Since NAND and fan-out are realizable with Toffoli, any logical function is realizable.

Quantum Toffoli:

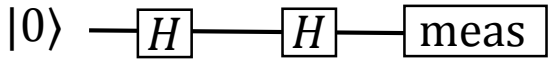


Importance of garbage collection

In classical reversible computing, a lot of garbage is a technical problem. There are some ways to reduce amount of garbage in the classical case, reducing garbage decreases the number of bits, which should be eventually erased (decreases the energy cost of $k_B T \ln 2$ per erased bit).

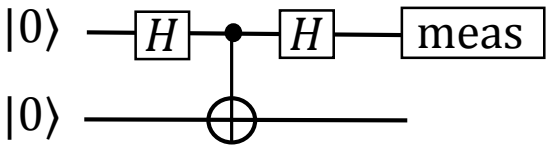
In QC collecting garbage is absolutely necessary.

Example



Result is always 0

Now "copy"



Result is 0 or 1 (50% - 50%)

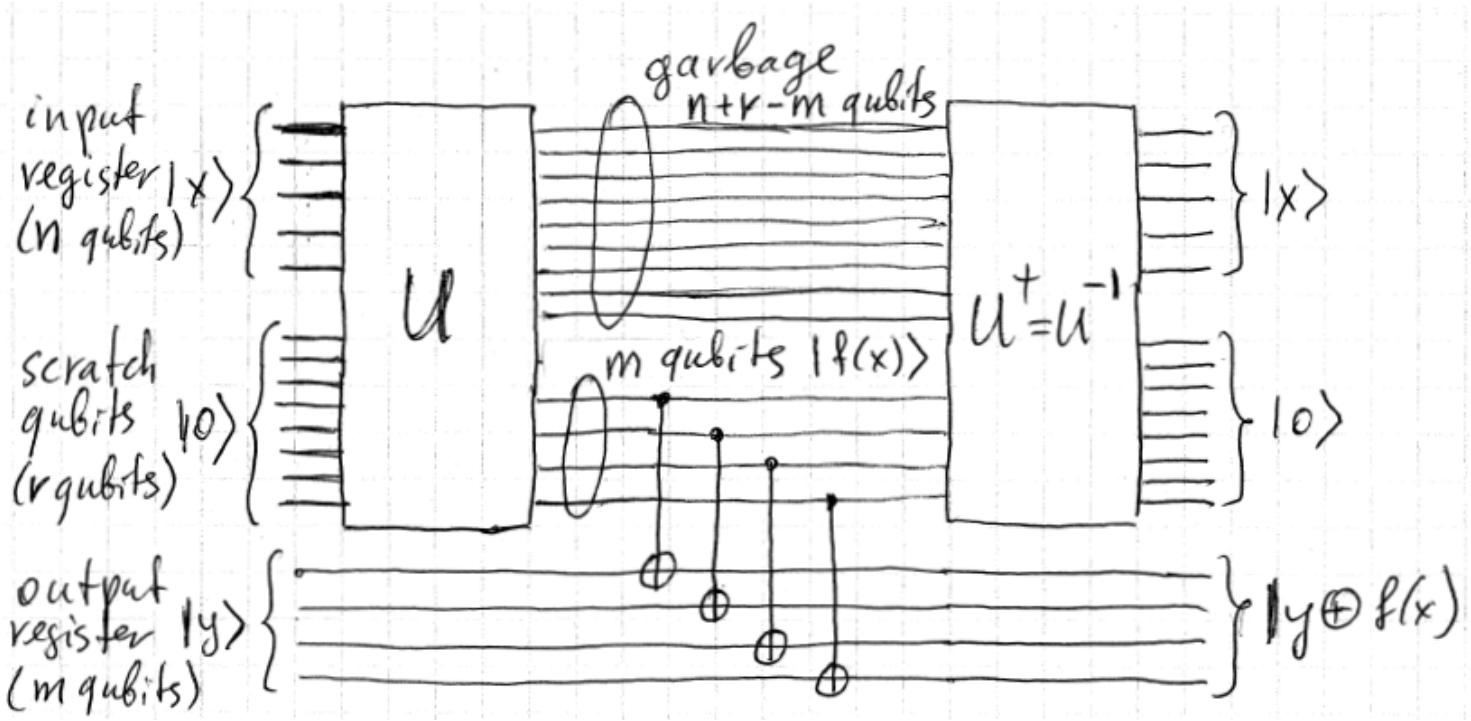
$$|00\rangle \xrightarrow{H} \frac{|0\rangle + |1\rangle}{\sqrt{2}}|0\rangle \xrightarrow{\text{CNOT}} \frac{|00\rangle + |11\rangle}{\sqrt{2}} \xrightarrow{H} \frac{(|0\rangle + |1\rangle)|0\rangle + (|0\rangle - |1\rangle)|1\rangle}{2} =$$

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

So, it is really important to get rid of garbage

Garbage collection procedure

In principle, any entanglement with “scratch” qubits can be erased by running computation backwards.

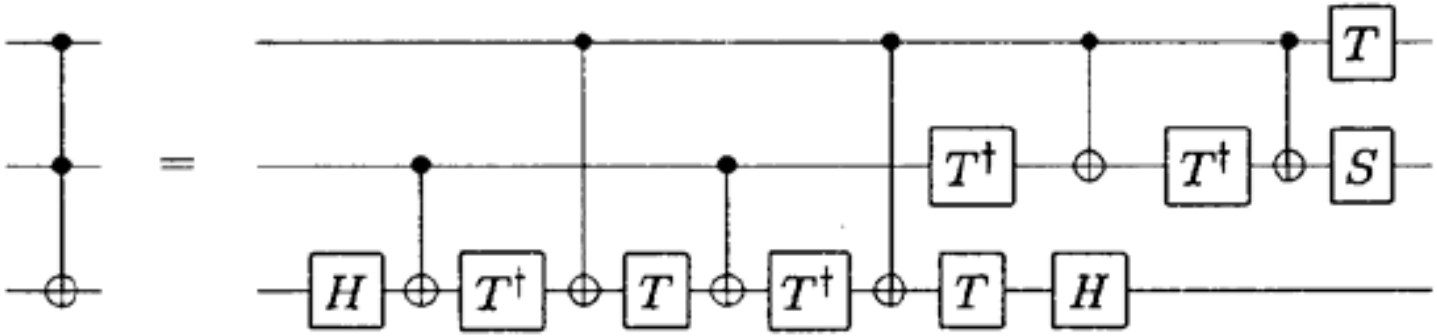


“copy”

Technically, U consists of gates; we need to use conjugate gates in reverse order. For many gates there is no need to conjugate: $X^\dagger = X, Y^\dagger = Y, Z^\dagger = Z, H^\dagger = H,$ $(\mathrm{CNOT})^\dagger = \mathrm{CNOT}$, $(\mathrm{Toffoli})^\dagger = \mathrm{Toffoli}$ (however, $S^\dagger \neq S, T^\dagger \neq T$).

Realization of Toffoli with CNOTs and 1-qubit gates

Standard circuit (requires 6 CNOTs)



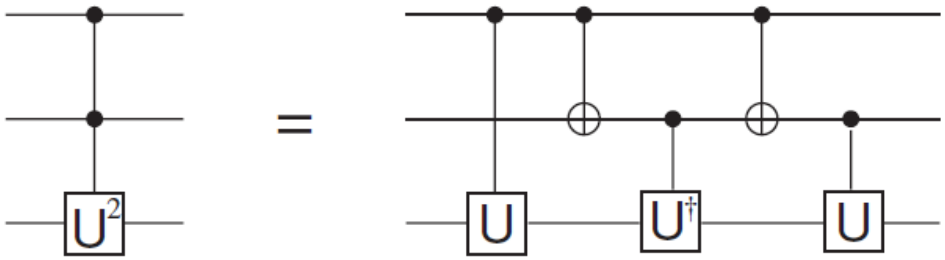
$$S = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix} \qquad T = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix}$$

Not easy to understand this equivalence,
but can be checked formally

In Mermin's book there are two other (understandable) constructions of Toffoli gate with CNOTs and single-qubit gates

Mermin's constructions for Toffoli gate

Lemma 1

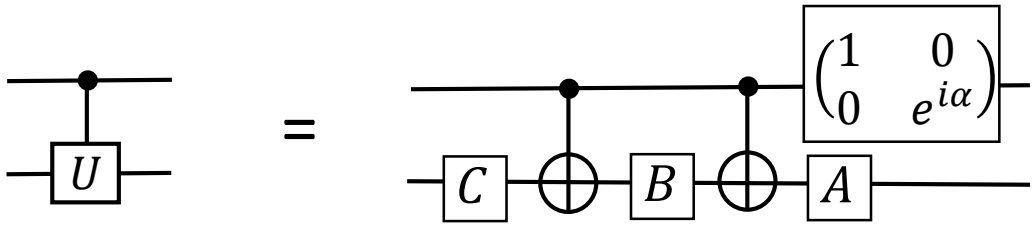


Straightforward proof: check states $|00\rangle$, $|01\rangle$, $|10\rangle$, and $|11\rangle$

For Toffoli we need $U = \sqrt{X} = H\sqrt{Z}\,H$ (the last equation follows from $X = HZH$)

(The gate controlled-$\sqrt{Z}$ can be naturally realized physically, then 5 two-qubit gates.)

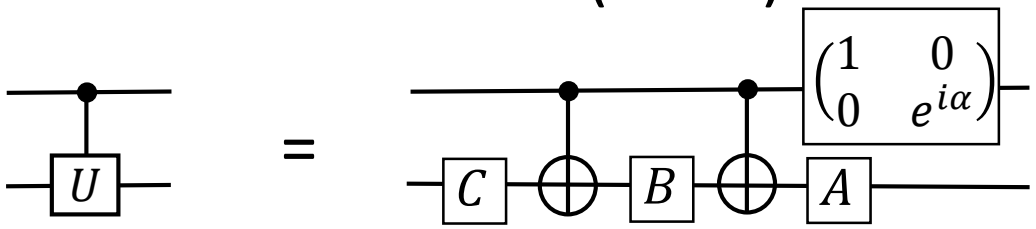
Lemma 2 For any unitary U, we can find unitary A, B, C, and real number α, so that



Then Toffoli can be realized with 8 CNOTs

Mermin's constructions for Toffoli (cont.)

Proof of Lemma 2:



First, the gate $\begin{pmatrix}1 & 0\\ 0 & e^{i\alpha}\end{pmatrix}$ is equivalent to controlled-$e^{i\alpha}\hat{1}$, so we need

$$\begin{cases} ABC = \hat{1} \\ e^{i\alpha} AXBXC = U \end{cases}$$

Proof by explicit construction: for $U = e^{i\alpha} R_Z(\beta)\, R_Y(\gamma)\, R_Z(\delta)$ (Euler angles β, γ, δ)

$e^{-i\frac{\beta}{2}Z}$ $e^{-i\frac{\gamma}{2}Y}$ $e^{-i\frac{\delta}{2}Z}$

We choose $A = R_Z(\beta)\, R_Y\left(\frac{\gamma}{2}\right)$, $B = R_Y\left(-\frac{\gamma}{2}\right) R_Z\left(-\frac{\delta+\beta}{2}\right)$, $C = R_Z\left(\frac{\delta-\beta}{2}\right)$

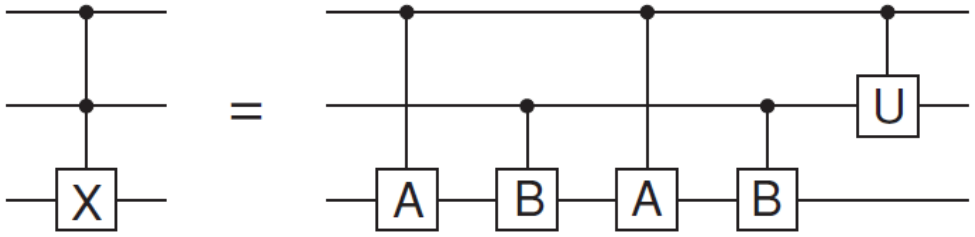
Then $ABC = R_Z(\beta) \underbrace{R_Y\left(\frac{\gamma}{2}\right) R_Y\left(-\frac{\gamma}{2}\right)}_{\hat{1}} \underbrace{R_Z\left(-\frac{\delta+\beta}{2}\right) R_Z\left(\frac{\delta-\beta}{2}\right)}_{R_Z(-\beta)} = \hat{1}$

$$AXBXC = R_Z(\beta)\, R_Y\left(\frac{\gamma}{2}\right) \underbrace{X\, R_Y\left(-\frac{\gamma}{2}\right)}_{R_Y(+\gamma/2)} \overbrace{X\hat{1}X}^{\hat{1}} \underbrace{R_Z\left(-\frac{\delta+\beta}{2}\right) X}_{R_Z(+(\delta+\beta)/2)} R_Z\left(\frac{\delta-\beta}{2}\right) =$$

$= R_Z(\beta)\, R_Y(\gamma)\, R_Z(\delta)$ QED "spin echo"

Mermin's constructions for Toffoli (cont.)

Another construction
(with 6 CNOTs)



where $U = S^\dagger = \begin{pmatrix} 1 & 0 \\ 0 & -i \end{pmatrix}$, and we need $\begin{cases} A^2 = B^2 = \hat{1} \\ BABA = iX \end{cases}$

Since $A^2 = B^2 = \hat{1}$, they are kind of similar to X, and therefore c-A and c-B require only 1 CNOT (not two CNOTs)

Choose $A = \vec{a}\vec{\sigma}$, $B = \vec{b}\vec{\sigma}$, where $\vec{\sigma}$ are Pauli matrices and $\vec{a}, \vec{b}$ are unit vectors (directions). Then $BA = \left(\vec{b}\vec{\sigma}\right)(\vec{a}\vec{\sigma}) = \vec{b}\vec{a}\,\hat{1} + i\left(\vec{b} \times \vec{a}\right)\vec{\sigma}$.

vector product

So, we can pick $\vec{a}$ and $\vec{b}$ in yz plane, with angle $\pi/4$ between them.
Then $\vec{b} \times \vec{a}$ is along x-axis, and $BA = \cos(\pi/4)\,\hat{1} + i\sin(\pi/4)\,\sigma_x$,
therefore $(BA)^2 = \cos(\pi/2)\;\hat{1} + i\sin(\pi/2)\,\sigma_x = iX$, as needed.