Any $f(x)$ can be computed reversibly:

$$f^R : (x, y) \longmapsto (x, f(x) \oplus y)$$

($\oplus$ ← bitwise XOR.)

- possibly w/ ancillas – w/out changing anything else.

(Idea: Compute f reversibly w/ ancillas, XOR result w/ y, and "un-compute" everything – clean ancillas.
– Can be optimized to use few ancillas! → Prakash)

⇒ Everything can be computed reversibly.

But: 3-bit gate needed (→ HW)

5) Quantum Circuits

Model of quantum computation:

- System consists of qubits (or d-its): tensor prod. structure
- Choose universal gate set

$$S = \{U_1, \ldots, U_k\} \text{ of "small" (i.e. few-qubit) gates}$$

⇒ Small (poly-size) circuits.

- Input: Classical input $|x_1\rangle|x_2\rangle \dots |x_n\rangle$

  in computational basis.

- "Output": Measure all qubits at the end in the "computational basis" $\{|0\rangle, |1\rangle\}$.

  (Note: Other measurements do not help: can use unit. gate set to do any POVM.

  - Not meas. gubits = tracing out = meas. and ignoring result.
  - Meas. at earlier time: can be postponed until end (no difference!)

)

Which gate set should we choose?

- Continuum of gates: much more rich!
- turns out: 1+2 qubit gate sufficient (unlike classical!)
- Almost all 2-qubit-gates are universal by themselves (in an appropr. sense, if involved phases incommensurate)

Our choice:

(i) 1-qubit rotations about x & z axis:

$$R_x(\phi) = e^{-i X \phi/2} \quad ; \quad X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \; ; \; X^2 = I$$

$$R_z(\phi) = e^{-i Z \phi/2} \quad ; \quad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \; ; \; Z^2 = I$$

For $\pi^2 = I$: $e^{i\pi\phi/2} = \cos\phi/2 \, I + i \sin\phi/2 \, \pi$

$$\rightarrow R_x(\phi) = \begin{pmatrix} \cos\phi/2 & -i\sin\phi/2 \\ -i\sin\phi/2 & \cos\phi/2 \end{pmatrix};$$

$$R_z(\phi) = \begin{pmatrix} e^{-i\phi/2} & 0 \\ 0 & e^{+i\phi/2} \end{pmatrix}$$

$\Rightarrow$ Can generate any rotation $U \in SU(2)/\mathbb{Z}_2 \cong SO(3)$
(→ Euler angles!)

$$U = R_x(\alpha) R_z(\beta) R_x(\gamma)$$

(ii) one 2-qubit gate (almost all are ok...)

Typ. we choose "controlled-NOT"

$$CNOT = \quad x \to x, \quad y \to x \oplus y \quad = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

$\Rightarrow$ flips y iff $x=1$: Classical gate!

Turns out: Univ. gate set can be used to exactly create any n-qubit gate. (Of course not eff.: there are "even more" n-qubit unitaries than n-bit functions.)

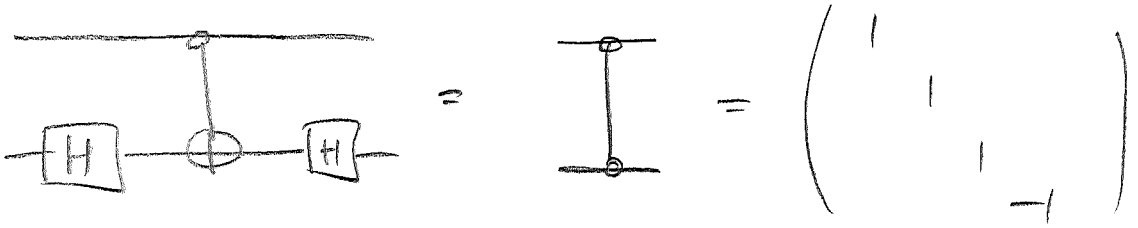
Some important gate sets + identities: ($\rightarrow$ HW):

Hadamard gate: $H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$; $H = H^\dagger$

$$H R_x(\phi) H = R_z(\phi)$$

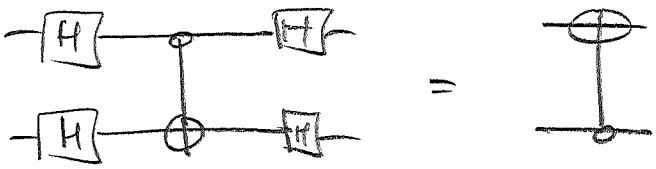
$$H R_z(\phi) H = R_x(\phi)$$

Graphical notation:

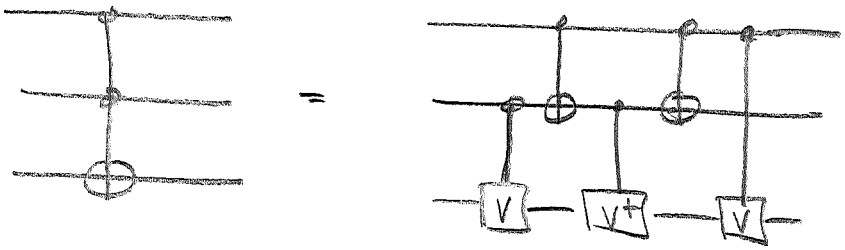


"Controlled-Z"

"Controlled-Phase", CPHASE



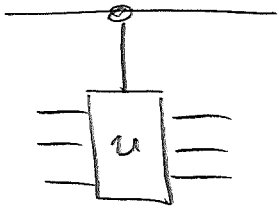
Toffoli:



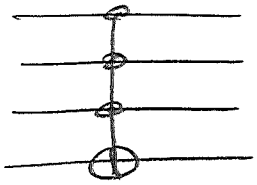
with $V = \frac{1-i}{2}(I + iX)$, and  = $\left(\begin{array}{c|c} 1 & 0 \\ \hline 0 & V \end{array}\right)$

"controlled -V"

Moreover: If we know how to build any classical U, we can also build "controlled - U":



→ just replace every Toffoli (class. univ.!) by Toffoli w/ 3 controls:



: can be built from normal Toffoli (→ HW)

What are other univ. gate sets? → Many!

Note: Different notions of universality:

- exact univ.: any n-qubit U can be realized exactly
- approx. univ.: any n-qubit U can be approx. by gate set (w/ reasonable conv.?)

Examples of approx. univ. gate sets:

- CNOT + 2 random 1-qubit gates
- CNOT + H + $T = R_z(\pi/4)$ ("$\pi/8$-gate")
- most 2-qubit gates alone

Solovay-Kitaev-Thm: A universal gate set for $SU(2^n)$ can approximate any $U \in SU(2^n)$ up to ε with $O(\text{poly}(\log(1/\varepsilon)))$ gates.

2. Oracle-based algorithms

a) The Deutsch algorithm

Consider $f: \{0,1\} \to \{0,1\}$.

Let f be "very hard" to compute (e.g. by circuit).

Want to know: Is $f(0) \stackrel{?}{=} f(1)$?

How often do we have to evaluate f?

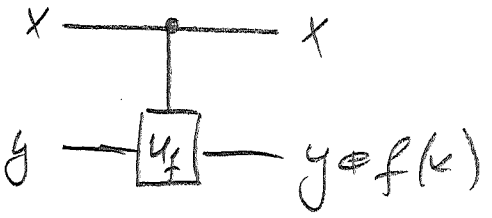
(Treat f as "black box" = "oracle": how many queries to oracle?)

→ Classically: 2 queries: $f(0)$, $f(1)$.

Can Q.M. help?

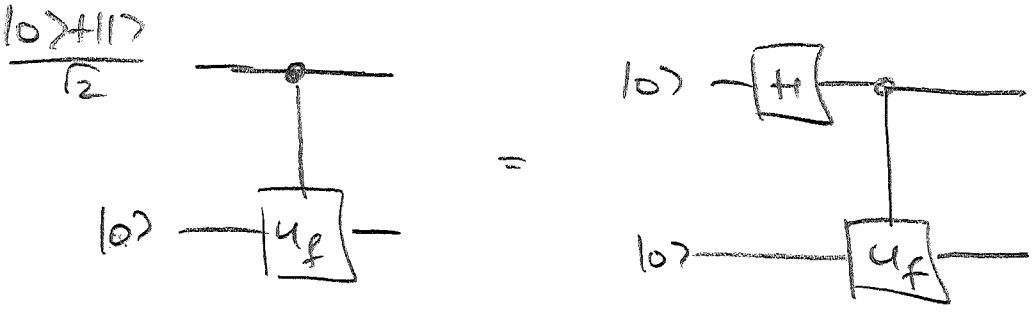
Consider reversible implementation of f:

$$f^R : (x,y) \longmapsto (x, y \oplus f(x))$$



$: |x\rangle|y\rangle \longmapsto |x\rangle|y \oplus f(x)\rangle$

→ try to use superpositions?



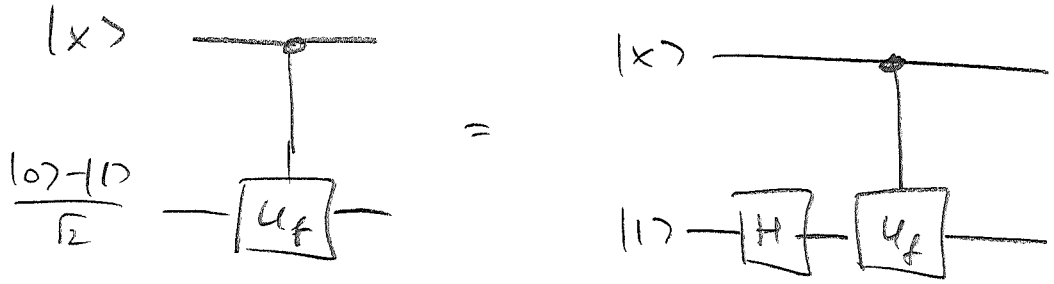
$$\left(\frac{|0\rangle+|1\rangle}{\sqrt{2}}\right)|0\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle|0\rangle + |1\rangle|0\rangle\right) \xrightarrow{U_f} \frac{1}{\sqrt{2}}\left(|0\rangle|f(0)\rangle + |1\rangle|f(1)\rangle\right)$$

$\Rightarrow$ Have evaluated f on both inputs!

But: how can we extract (relevant) information?

- Meas. of qubit 1: collapse superposition!
- Meas. of qubit 2: ?

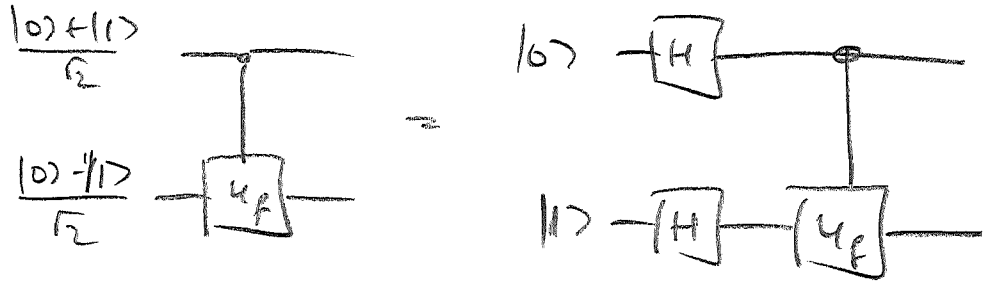
Consider



$$|x\rangle\left(\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right) \longmapsto |x\rangle\left(\frac{|f(x)\rangle - |1\oplus f(x)\rangle}{\sqrt{2}}\right) =$$

$$= \left\{\begin{array}{ll} \underline{f(x)=0}: & |x\rangle\,\dfrac{|0\rangle-|1\rangle}{\sqrt{2}} \\ \underline{f(x)=1}: & |x\rangle\,\dfrac{|1\rangle-|0\rangle}{\sqrt{2}} \end{array}\right\} = |x\rangle\cdot\left[(-1)^{f(x)}\,\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right].$$

Combine:



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

→ no entanglement created (!)

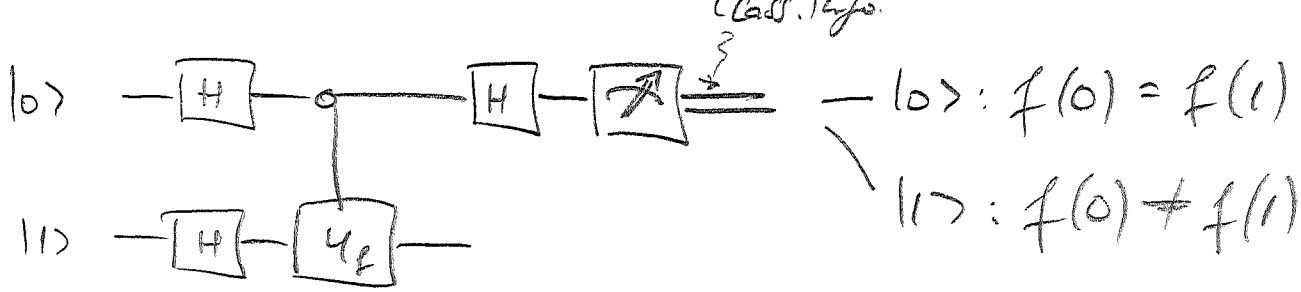
→ 2nd qubit unchanged (!!)

→ 1st qubit gets phase $(-1)^{f(x)}$

⇒ "phase kick-back" technique

⇒ 1st qubit = $\frac{|0\rangle+|1\rangle}{\sqrt{2}}$: $f(0)=f(1)$

$= \frac{|0\rangle-|1\rangle}{\sqrt{2}}$: $f(0)\neq f(1)$.



⇒ One application of U_f sufficient!

→ factor 2 faster than classically!

Note: 2nd qubit need not be measured (and contains no information!)

Two core ideas:

- Use input $\sum|x\rangle$ to evaluate f on all inputs simultaneously
- Need way to read out relevant info!

6) The Deutsch-Jozsa algorithm

Consider $f: \{0,1\}^n \to \{0,1\}$ w/ promise that

either $f(x) = c \ \forall x$ "constant"

or $\#\{x \mid f(x)=0\} = \#\{x \mid f(x)=1\}$ "balanced"

Want to know: Is f constant or balanced?

→ How many queries to f do we need?

Use same idea: Input $\sum|x\rangle$ and $\frac{|0\rangle-|1\rangle}{\sqrt{2}}$.