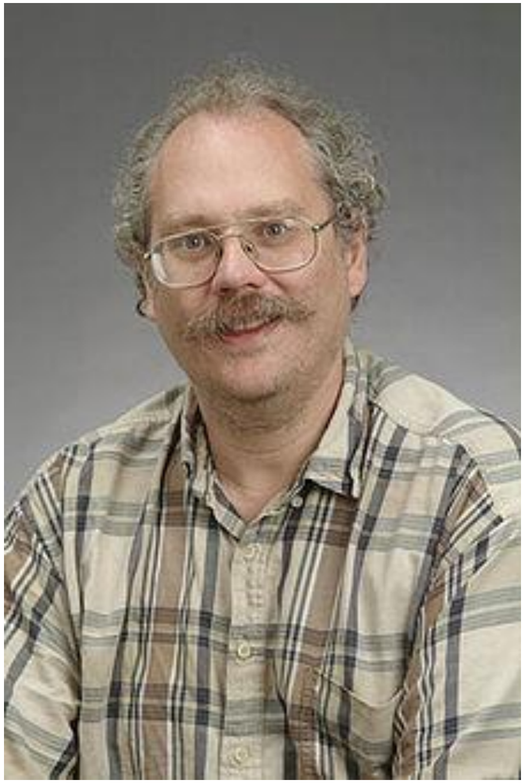
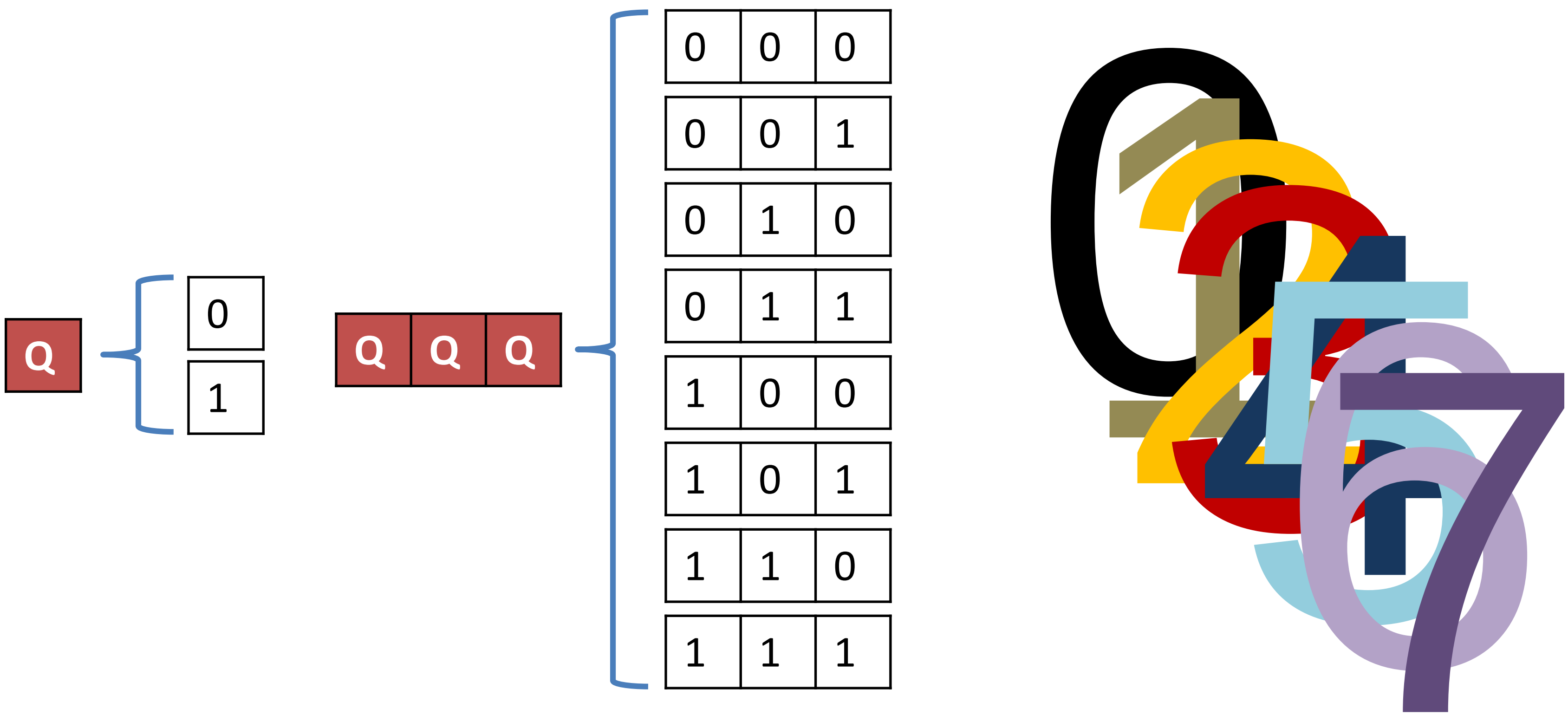


Peter Shor

Kvantni algoritmi

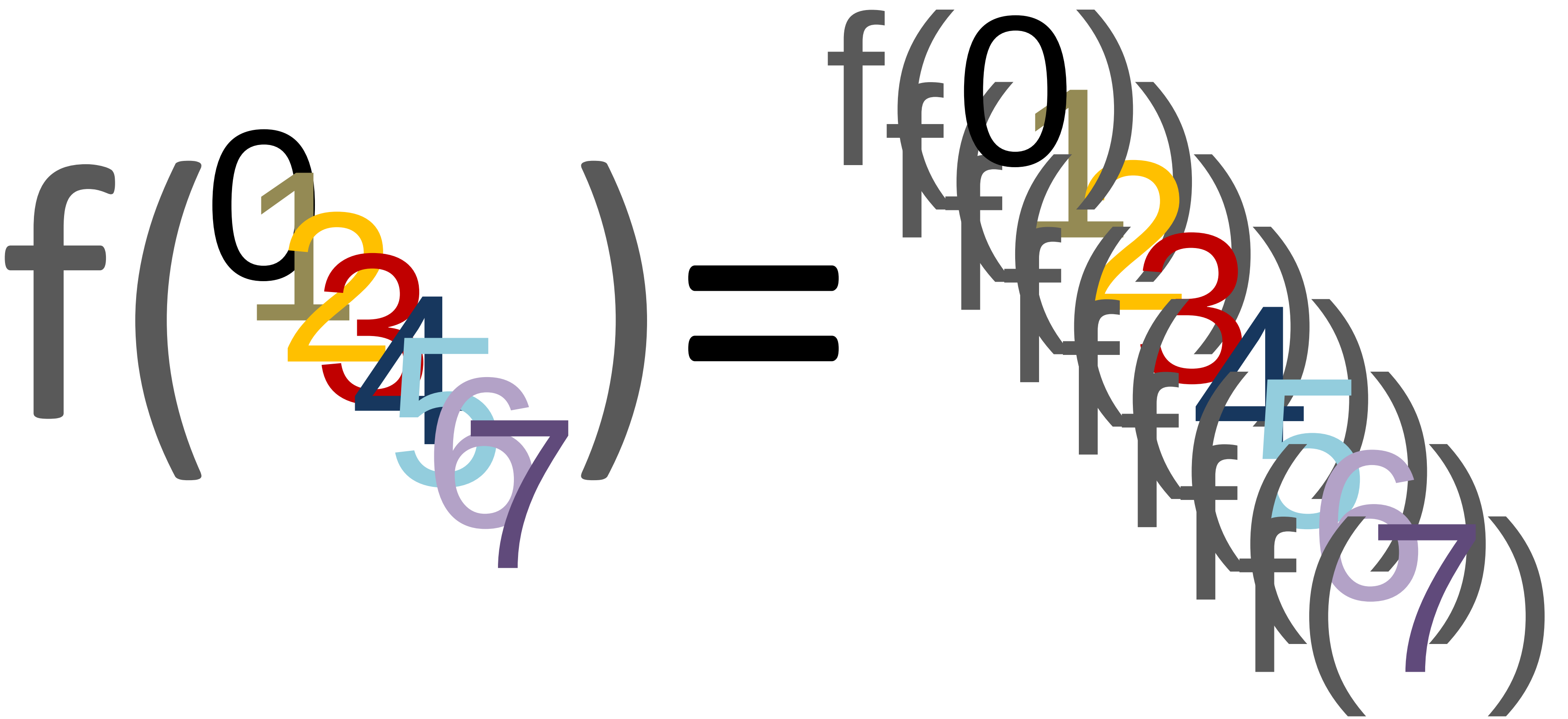
Vir: http://www-math.mit.edu/~shor/

Kvantni register



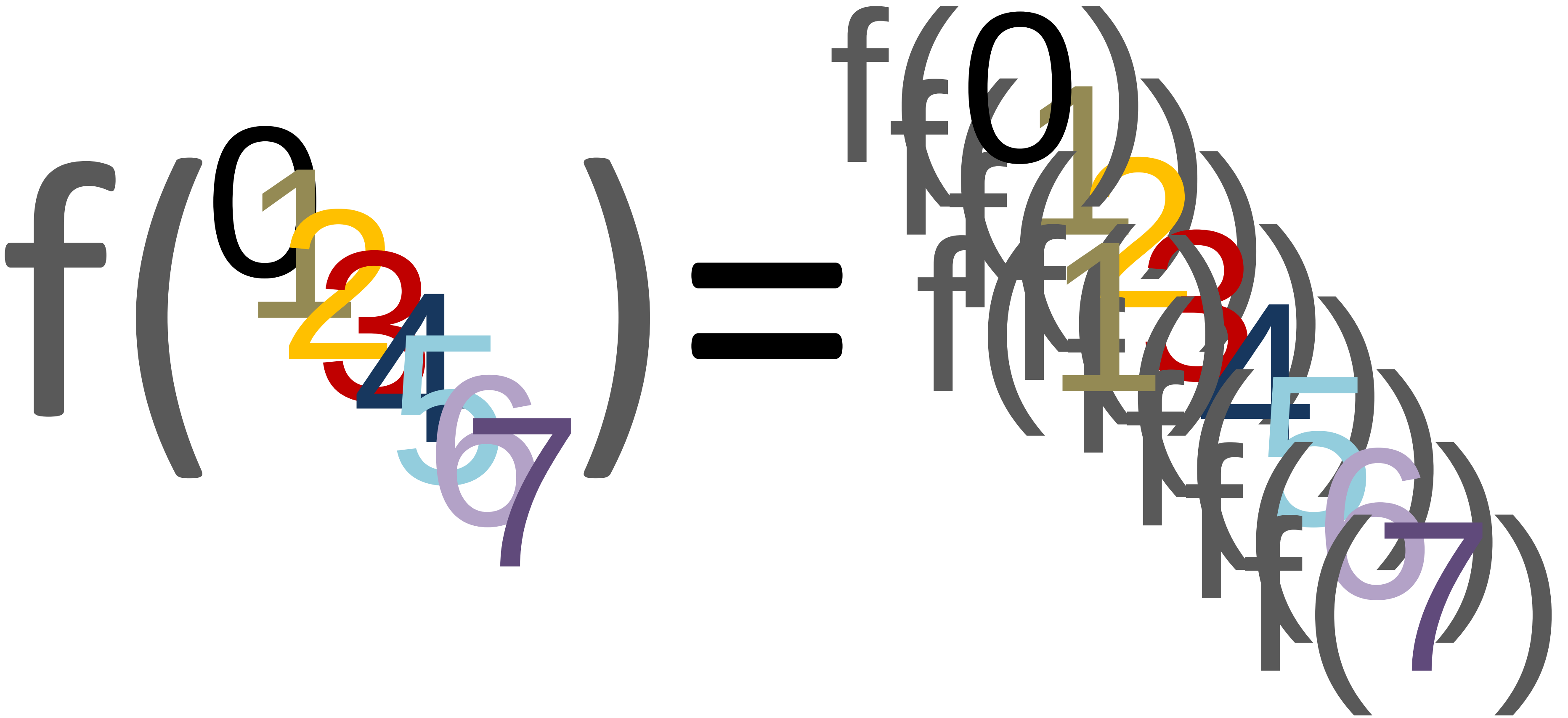
100 kvantnih bitov lahko hrani več klasičnih bitov informacij kot je atomov v vidnem vesolju!

Kvantni register in funkcije



Pripravi Razvij Izmeri

Kvantni register in funkcije



Pripravi Razvij Izmeri

Kvantno računalništvo

Kvantno stanje z n kvantnimi biti potrebuje 2^n kompleksnih števil za opis stanja:

$$\sum_{x \in \{0,1\}^n} \alpha_x |x\rangle$$

Cilj kvantnega računalništva je izkoristiti to superpozicijo eksponentno mnogo stanj v izračunih in s tem algoritme, ki imajo eksponentno časovno zahtevnost izračunati v polinomskem času.

Ideja: Amplitude verjetnosti moramo nastaviti tako, da bodo poti, ki vodijo do nepravilnih odgovorov interferirale destruktivno in se s tem izničile, poti, ki vodijo do pravilnih odgovorov pa bodo interferirale konstruktivno.

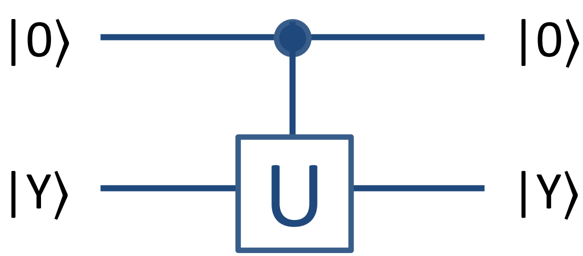
Kvantna vrata: Controlled-U

- vrata nad dvema kvantnima bitoma, ki uporabijo unitarno operacijo (matriko) **U** nad drugim kvantnim bitom, a samo če je prvi, kontrolni (prvi) kvantni bit postavljen na 1.

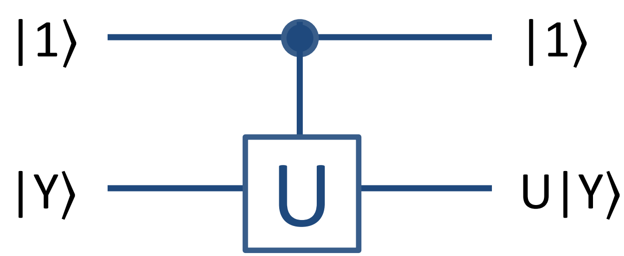
$$Controlled-U = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & u_{11} & u_{12} \\ 0 & 0 & u_{21} & u_{22} \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \\ \gamma \\ \delta \end{bmatrix} \left.\begin{matrix} |00\rangle \\ |01\rangle \\ |10\rangle \\ |11\rangle \end{matrix}\right\} \text{baza prostora}$$

kontrolni bit drugi bit

kontrolni kv. bit postavljen na 0



kontrolni kv. bit postavljen na 1



$$\mathbf{U} = \begin{bmatrix} u_{11} & u_{12} \\ u_{21} & u_{22} \end{bmatrix}$$

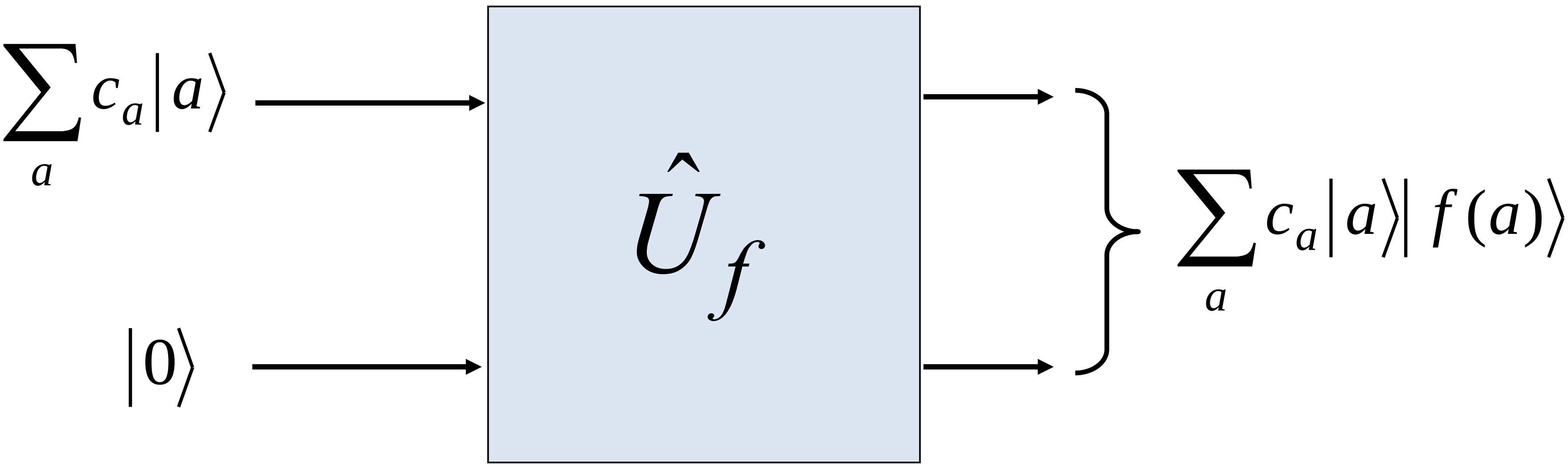
Unitarne transformacije

- Za katerokoli Boolovo funkcijo *f*: $\{0,1\}^n \rightarrow \{0,1\}$ obstaja unitarna transformacija kvantnega stanja

$$|x\rangle|0\rangle \rightarrow |x\rangle|f(x)\rangle$$

- Toda večino funkcij *f* ne moremo implementirati **učinkovito.** Zato nas trenutno zanimajo le tiste funkcije *f*, ki jih lahko sestavimo iz relativno majhnega števila kvantnih vrat (glede na velikost vhodih podatkov *n*).

Unitarne transformacije in kvantni registri



- Namesto vhodnega (kontrolnega) in izhodnega (drugega) kv. bita lahko imamo celotne kvantne registre.
- Če je vhodni register v superpoziciji več bitnih zaporedij (bitnih nizov) *a*, je izhodni register v superpoziciji (kvantni entangulaciji) vrednosti *f*(*a*) (po ena vrednost *f*(*a*) za vsako vhodno vrednost *a)*.

http://www.quantiki.org/wiki/Basic_concepts_in_quantum_computation

Deutsch-ov algoritem

Črna škatla izračuna eno izmed štirih možnih enobitnih funkcij:

Konstantna funkciji:

$$\boxed{\begin{matrix} f(0)=0 \\ f(1)=0 \end{matrix}}\ ali\ \boxed{\begin{matrix} f(0)=1 \\ f(1)=1 \end{matrix}}$$

Uravnoteženi funkciji:

$$\boxed{\begin{matrix} f(0)=0 \\ f(1)=1 \end{matrix}}\ ali\ \boxed{\begin{matrix} f(0)=1 \\ f(1)=0 \end{matrix}}$$

Radi bi vedeli, ali je naša črna škatla konstantna ali uravnotežena.
To lahko vedno ugotovimo z dvema izračunoma: $f\,(0)$ in $f\,(1)$.

Ali lahko to ugotovimo z enim samim izračunom?

- skonstruirajmo funkcijo: $|x\rangle|y\rangle \rightarrow |x\rangle|f(x) \oplus y\rangle$
- če **f(0)=f(1)=0**, potem
  - $|0\rangle|0\rangle \rightarrow |0\rangle|0\rangle$
  - $|0\rangle|1\rangle \rightarrow |0\rangle|1\rangle$
  - $|1\rangle|0\rangle \rightarrow |1\rangle|0\rangle$
  - $|1\rangle|1\rangle \rightarrow |1\rangle|1\rangle$

$$U_f = \begin{array}{cc} & \text{baza prostora} \\ \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} & \begin{array}{c} |00\rangle \\ |01\rangle \\ |10\rangle \\ |11\rangle \end{array} \end{array}$$

- če **f(0)=f(1)=1**, potem
  - $|0\rangle|0\rangle \rightarrow |0\rangle|1\rangle$
  - $|0\rangle|1\rangle \rightarrow |0\rangle|0\rangle$
  - $|1\rangle|0\rangle \rightarrow |1\rangle|1\rangle$
  - $|1\rangle|1\rangle \rightarrow |1\rangle|0\rangle$

$$U_f = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{array}{c} |00\rangle \\ |01\rangle \\ |10\rangle \\ |11\rangle \end{array}$$

vir: http://web.stanford.edu/~rsasaki/AP227/chap5.pdf

- skonstruirajmo funkcijo: $|x\rangle|y\rangle \rightarrow |x\rangle|f(x) \oplus y\rangle$
- če **f(0)=0, f(1)=1**, potem
  - $|0\rangle|0\rangle \rightarrow |0\rangle|0\rangle$
  - $|0\rangle|1\rangle \rightarrow |0\rangle|1\rangle$
  - $|1\rangle|0\rangle \rightarrow |1\rangle|1\rangle$
  - $|1\rangle|1\rangle \rightarrow |1\rangle|0\rangle$

baza prostora

$$U_f = \begin{array}{cc} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} & \begin{array}{c} |00\rangle \\ |01\rangle \\ |10\rangle \\ |11\rangle \end{array} \end{array}$$

- če **f(0)=1, f(1)=0**, potem
  - $|0\rangle|0\rangle \rightarrow |0\rangle|1\rangle$
  - $|0\rangle|1\rangle \rightarrow |0\rangle|0\rangle$
  - $|1\rangle|0\rangle \rightarrow |1\rangle|0\rangle$
  - $|1\rangle|1\rangle \rightarrow |1\rangle|1\rangle$

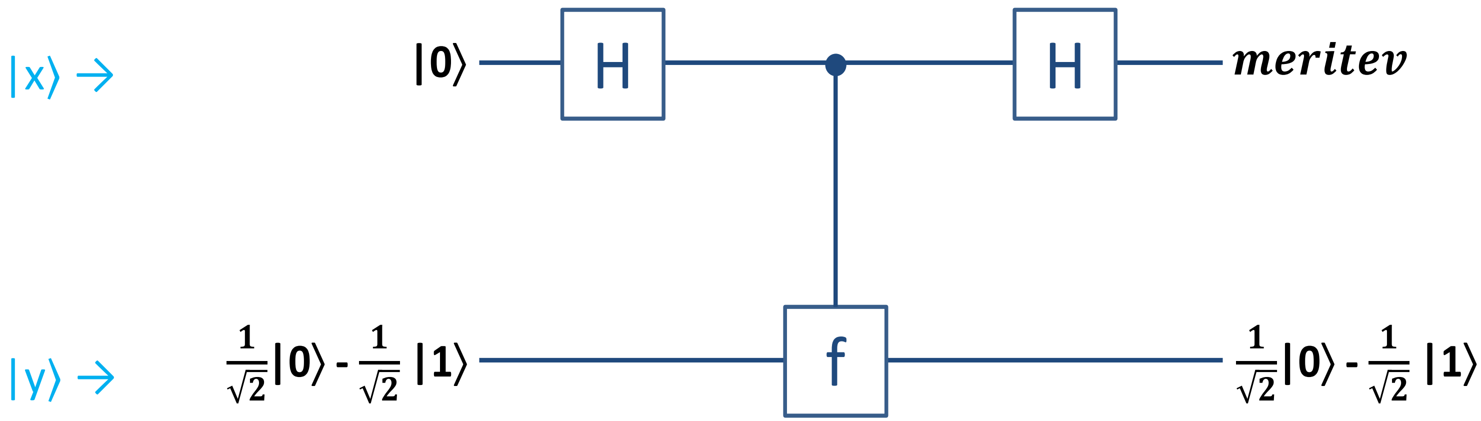
$$U_f = \begin{array}{cc} \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} & \begin{array}{c} |00\rangle \\ |01\rangle \\ |10\rangle \\ |11\rangle \end{array} \end{array}$$

vir: http://web.stanford.edu/~rsasaki/AP227/chap5.pdf

Deutsch-ov algoritem

$$\mathbf{H}=\frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}$$

- Odgovor z eno samo evalvacijo funkcije f dobimo s pomočjo naslednjega kvantnega algoritma:



- Po prvih Hadamardovih vratih je stanje obeh kv. bitov (če izpustimo normalizacijo s √2) [1]:

$$\underbrace{(\;|0\rangle+|1\rangle\;)}_{|x\rangle}\underbrace{(|0\rangle - |1\rangle\;)}_{|y\rangle}$$

[1] A. Ekert, P. Hayden H. Inamori: Basic concepts in quantum computation, 2008

- Po prvih Hadamardovih vratih **H** je stanje: $\underbrace{(\ |0\rangle+|1\rangle\ )}_{|x\rangle}\underbrace{(|0\rangle - |1\rangle\ )}_{|y\rangle}$
- če **f(0)=f(1)=0**, potem

$$U_f = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} +1 \\ -1 \\ +1 \\ -1 \end{bmatrix} \begin{matrix} |00\rangle \\ |01\rangle \\ |10\rangle \\ |11\rangle \end{matrix} = \begin{bmatrix} +1 \\ -1 \\ +1 \\ -1 \end{bmatrix} \rightarrow \underbrace{(\ |0\rangle+|1\rangle\ )}_{|x\rangle}\underbrace{(|0\rangle - |1\rangle\ )}_{|y\rangle}$$

in po drugih vratih **H** imamo $\mathbf{H}x = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 2 \\ 0 \end{bmatrix} = |0\rangle$

- če **f(0)=f(1)=1**, potem

$$U_f = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} +1 \\ -1 \\ +1 \\ -1 \end{bmatrix} \begin{matrix} |00\rangle \\ |01\rangle \\ |10\rangle \\ |11\rangle \end{matrix} = \begin{bmatrix} -1 \\ +1 \\ -1 \\ +1 \end{bmatrix} \rightarrow \underbrace{(\ -|0\rangle-|1\rangle\ )}_{|x\rangle}\underbrace{(|0\rangle - |1\rangle\ )}_{|y\rangle}$$

in po drugih vratih **H** imamo $\mathbf{H}x = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} -1 \\ -1 \end{bmatrix} = \begin{bmatrix} -2 \\ 0 \end{bmatrix} = |0\rangle$

Tenzorski produkt

$$(\alpha|0\rangle+\beta|1\rangle) \otimes (\gamma|0\rangle+\delta|1\rangle) = \alpha\gamma|00\rangle+\alpha\delta|01\rangle+ \beta\gamma|10\rangle+\beta\delta|11\rangle$$

faktorizacija

ZGLEDI:

$$(1|0\rangle+1|1\rangle) (1|0\rangle - 1|1\rangle) = 1|00\rangle+ -1|01\rangle+ 1|10\rangle+ -1|11\rangle$$

$$(-1|0\rangle -1|1\rangle) (1|0\rangle - 1|1\rangle) = \underline{\quad}|00\rangle+ \underline{\quad}|01\rangle+ \underline{\quad}|10\rangle+ \underline{\quad}|11\rangle$$

- Po prvih Hadamardovih vratih **H** je stanje: $\underbrace{(\,|0\rangle+|1\rangle\,)}_{|x\rangle}\underbrace{(\,|0\rangle-|1\rangle\,)}_{|y\rangle}$
- če **f(0)=0, f(1)=1**, potem

$$U_f=\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&0&1\\0&0&1&0\end{bmatrix}\begin{bmatrix}+1\\-1\\+1\\-1\end{bmatrix}\begin{matrix}|00\rangle\\|01\rangle\\|10\rangle\\|11\rangle\end{matrix}=\begin{bmatrix}+1\\-1\\-1\\+1\end{bmatrix}\rightarrow\underbrace{(\,|0\rangle-|1\rangle\,)}_{|x\rangle}\underbrace{(\,|0\rangle-|1\rangle\,)}_{|y\rangle}$$

in po drugih vratih **H** imamo $\mathbf{H}x=\begin{bmatrix}1&1\\1&-1\end{bmatrix}\begin{bmatrix}1\\-1\end{bmatrix}=\begin{bmatrix}0\\2\end{bmatrix}=|1\rangle$

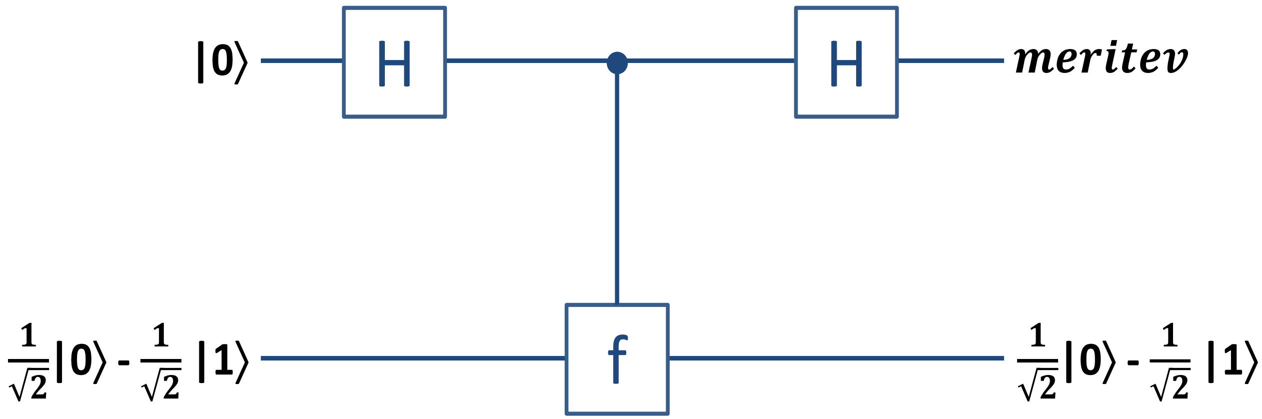
- če **f(0)=1, f(1)=0**, potem

$$U_f=\begin{bmatrix}0&1&0&0\\1&0&0&0\\0&0&1&0\\0&0&0&1\end{bmatrix}\begin{bmatrix}+1\\-1\\+1\\-1\end{bmatrix}\begin{matrix}|00\rangle\\|01\rangle\\|10\rangle\\|11\rangle\end{matrix}=\begin{bmatrix}-1\\+1\\+1\\-1\end{bmatrix}\rightarrow\underbrace{(\,-|0\rangle+|1\rangle\,)}_{|x\rangle}\underbrace{(\,|0\rangle-|1\rangle\,)}_{|y\rangle}$$

in po drugih vratih **H** imamo $\mathbf{H}x=\begin{bmatrix}1&1\\1&-1\end{bmatrix}\begin{bmatrix}-1\\+1\end{bmatrix}=\begin{bmatrix}0\\-2\end{bmatrix}=|1\rangle$

Deutsch-ov algoritem (krajše)

$$\mathbf{H} = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$



- po evalvaciji funkcije *f*, sta stanji [1]:

$$|x\rangle(|0\rangle - |1\rangle) \overset{f}{\longmapsto} (-1)^{f(x)}|x\rangle(|0\rangle - |1\rangle) =$$

$$[(-1)^{f(0)}|0\rangle + (-1)^{f(1)}|1\rangle](|0\rangle - |1\rangle)$$

- torej je prvi kv. bit($|x\rangle$) v stanju

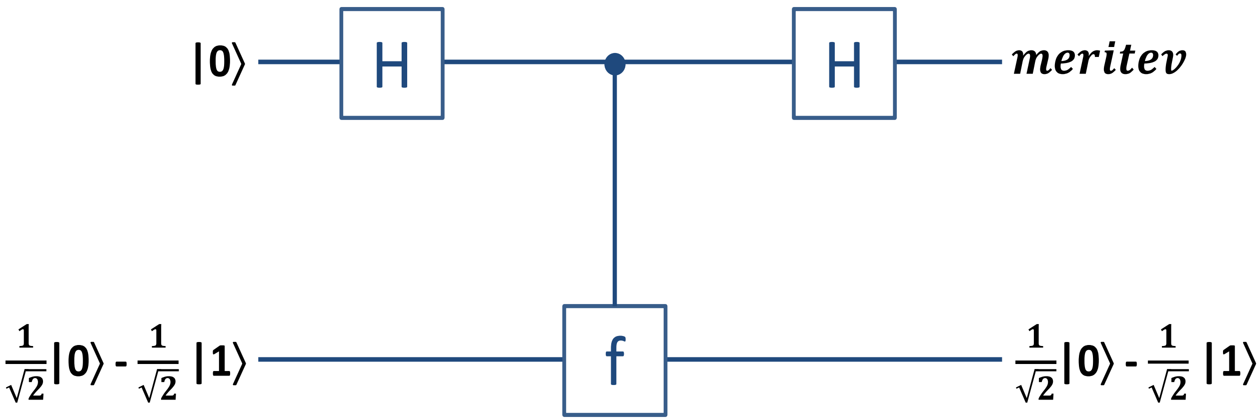
$$\pm(|0\rangle + |1\rangle), \text{ če } \; f(0) = f(1)$$

$$\pm(|0\rangle - |1\rangle), \text{ če } \; f(0) \neq f(1)$$

[1] A. Ekert, P. Hayden H. Inamori: Basic concepts in quantum computation, 2008

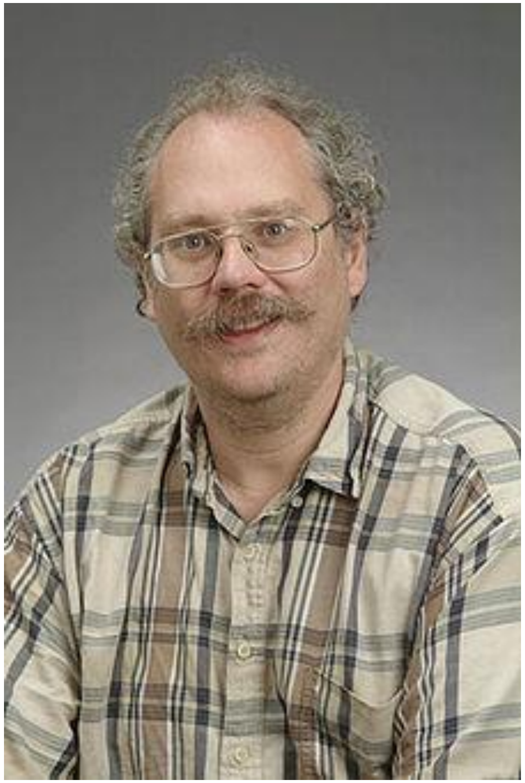
Deutsch-ov algoritem (krajše)

$$\mathbf{H} = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$



- Po drugih Hadamardovih vratih je stanje prvega kv. bita [1]:
  - $|0\rangle$ če je f **konstantna**
  - $|1\rangle$ če je f **uravnotežena**.

[1] A. Ekert, P. Hayden H. Inamori: Basic concepts in quantum computation, 2008



Shor-ov algoritem

Vir: http://www-math.mit.edu/~shor/

Shor-ov algoritem faktorizacije

Shor-ov kvantni algoritem za faktorizacijo velikih celih števil je daleč najbolj vpliven in slaven med vsemi kvantnimi algoritmi (1994).

Zaradi njega je močno poskočilo zanimanje in finančno vlaganje v razvoj kvantnih računalnikov saj omogoča učinkovito dekripcijo asimteričnih kodirnikov (RSA in diskretnih logaritmov).

Miljarde evrov so zaščitene s kriptografijo (vsi bančni sistemi, nepremičninski trgi, borze itd.)

Peter Shor je pokazal, kako lahko faktoriziramo velika števila v polinomskem času, za kar je na klasičnem računalniku potreben eksponenten čas. Shorov algoritem lahko faktorizira 2^N krat hitreje, kjer je N bitna velikost ključa.

Aritmetika po modulu: modularni inverz

- Celo število **a** ≥ 2 je praštevilo, če je deljivo samo z 1 in z **a**
- Največji skupni delitelj **d**=gcd(**a**,**b**) je največje celo število d, ki deli celi števili **a** in **b**.
- Celi števili **a** in **b** sta **tuji števili** če gdc(**a**,**b**) = 1;
- Za tuji celi števili **a** in **n** vedno obstaja unikatno število **d** ∈ {0, ..., n − 1} tako da velja

$$ad = 1 \bmod n$$

Število **d** imenujemo inverz števila **a** po modulu **n** in ga označimo z a^{-1}

Karl Friedrich Gauss (1777-1855)

Shor-ov algoritem: klasičen del

1. Izberi naključno število $a < N$
2. izračunaj gcd(a, N).
3. Če d=gcd(a, N) $\neq 1$, potem je d iskani netrivialni faktor N, torej smo končali.
4. V nasprotnem primeru uporabimo kvantno rutino za iskanje periode r funkcije:

   ,

   $$\boxed{f(x) = a^x \bmod N}$$

   r je red števila a v $(\mathbb{Z}_N)^\times$, torej najmanjše celo število, za katerega velja $f(x + r) = f(x)$
5. Če je r liho število, se vrni na korak 1.
6. Če $a^{r/2} \equiv -1 \pmod N$, se vrni na korak 1.
7. gcd($a^{r/2} \pm 1$, N) je netrivialni faktor N, torej smo končali.

Shor-ov algoritem: primer

Poskusimo faktorizirati število N = 15. Izberimo a=8 (8 in 15 sta tuji števili).
Torej imamo $f_{15}(x) = 8^x \bmod 15$
Za x = 0, 1, 2, … imamo ciklični vzorec

$$f_{15}(0) = 1, \quad f_{15}(1) = 8, \quad f_{15}(2) = 4, \quad f_{15}(3) = 2,$$
$$f_{15}(4) = 1, \quad f_{15}(5) = 8, f_{15}(6) = 4, \ldots$$

Vidimo, da je vzorec res cikličen 1,8,4,2,1,8,4,2,1,8,4,2… s periodo r = 4.

Izračunamo **$d=\gcd(a^{r/2}-1, N) = \gcd(63, 15) = 3$**. Drugi faktor (5) lahko najdemo z deljenjem (N/d).

Poskusimo faktorizirati še število N = 85. Izberimo a=31 (31 in 85 sta tuji števili). Torej imamo $f_{85}(x) = 31^x \bmod 85$, ki za izbrane x=0,1,2,… tvori ciklični vzorec 1, 31, 26, 41, 81, 46, 66, 6, 16, 71, 76, 61, 21, 56, 36,11, 1, 31,…
Perioda r=16 in **$d=\gcd(a^{r/2}-1, N) = 5$.**

Preizkusite še sami razne vrednosti N in a in se prepričajte, da postopek res deluje.

Srce Shor-ovega algoritma je iskanje periode r s pomočjo kvantne funkcije.
Ko najdemo r, je faktorizacija N preprosta.

Shor-ov algoritem: dokaz klasičnega dela

Po definiciji periode *r* imamo $f(r) = a^r \bmod N = 1$. Torej N deli a^r-1. Po koraku 5 imamo takšen a, da je $gcd(a, N) = 1$ in *r* sodo število.

Definirajmo b= $a^{r/2} \bmod N$. Torej je b kvadratni koren števila 1 po $mod\ N$. Velja $b \neq 1$, saj je po definiciji perioda funkcije $f(x)$ enaka r in ne $r/2$. Korak 6 zagotavlja tudi $b \neq -1$.

Trdimo, da je $d = gcd(b\text{-}1,N)$ netrivialen faktor števila N (torej $d \neq 1$ *in* $d \neq N$).

1. Ker velja $d < b\text{-}1 < N$, velja tudi $d \neq N$

2. Če bi veljalo $d = gcd(b\text{-}1,N)=1$, potem bi po Bezoutovi enakosti (poimenovani po francoskem matematiku Étiennu Bézoutu) obstajala takšni celi števili u in v, da bi veljalo

   $$(b\text{-}1)u+Nv = 1$$

   Ko pomnožimo obe strani zgornje enačbe z $(b+1)$, dobimo:

   $$(b^2\text{-}1)u+N(b+1)v = b+1$$

   Ker N deli $b^2\text{-}1 = a^r\text{-}1$, bi moral glede na zgornjo enačbo N deliti tudi $(b+1)$, torej bi veljalo $b \bmod N = \text{-}1$, kar je v nasprotju s korakom 6.

Torej je $d = gcd(b\text{-}1,N)$ res netrivialen faktor števila N.

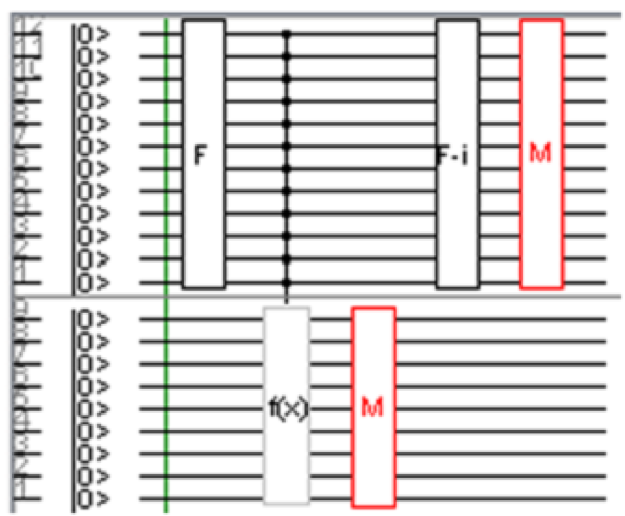
Opomba: Zgornji dokaz temelji na predpostavki, da obstaja takšno število b= $a^{r/2} \bmod N$, da $b \neq -1$ in $b \neq 1$. Obstoj takšnega števila b zagotavlja Teorem kitajskih ostankov, saj je N=pq sestavljeno iz praštevil.

Shor-ov algoritem: Kvantni del

Opis: Algoritem poišče periodo funkcije $f(x) = a^x \; mod \; N$, kjer je a poljubno število, ki je N tuje (gdc(a, N)=1), N pa je sestavljeno število: $N = pq$, kjer sta p in q praštevili.

Potrebna strojna oprema:

- vhodni kvantni register takšne velikosti Q, da je vanj možno hraniti število N^2.
- izhodni kvantni register takšne velikosti P, da je vanj možno hraniti število N.
- Fourierova kvantna vrata
- Hadamardova kvantna vrata
- kvantno vezje, ki implementira funkcijo $f(x) = a^x \; mod \; N$ (za vsak a in za vsak N potrebujemo posebno vezje).



vir slike: http://jquantum.sourceforge.net/

Slika: vezje kvantnega dela Shorovega algoritma z vhodnim registrom velikosti 12 qubitov in izhodnim registrom velikosti 9 qubitov.

Shor-ov algoritem: Kvantni del

Koraki algoritma:

1. INICIALIZACIJA:
   - vhodni kvantni register je v stanju 0
   - izhodni kvantni register v stanju 0

2. SUPERPOZICIJA VHODNEGA REGISTRA:
   - preko Hadamardove transformacije ali pa kvantne Fourierove transformacije postavimo vhodni kvantni register v popolno superpozicijo vseh možnih stanj:

$$\sum_x \frac{1}{Q} |x\rangle$$

   - izhodni kvantni register je še vedno v stanju 0

Shor-ov algoritem: Kvantni del

3. APLICIRANJE KVANTNE FUNKCIJE f(x):
   – vhodni kvantni register je še vedno v stanju $\sum_x \frac{1}{Q}|x\rangle$
   – izhodni kvantni register je v stanju $f\left(\sum_x \frac{1}{Q}|x\rangle\right)=\frac{1}{Q}\sum_x f(|x\rangle)$. Ker ima funkcija periodo r, zavzame samo r različnih vrednosti. Vse so enakovredno zastopane v izhodnem registru.

4. MERITEV IZHODNEGA REGISTRA:
   – izhodni kvantni register kolapsira v eno samo opazovano vrednost $y_0=f(x_0)$ (eno izmed tistih, ki so bile prej v superpoziciji izhodnega registra).
   – vhodni register posledično kolapsira v superpozicijo vseh tistih vhodov x_r, za katere velja $y_0=f(x_r)$. Ker je $f(x)$ periodična funkcija s periodo r, lahko to superpozicijo vhodnega registra zapišemo kot:

$$\frac{1}{Q}\sum_b |x_0+b\cdot r\rangle$$

kjer je b celo število, ki teče od 0 dokler x_0+rb ne preseže velikosti vhodnega registra Q.

Shor-ov algoritem: Kvantni del

5. INVERZNA KVANTNA FOURIEROVO TRANSFORMACIJA VHODNEGA REGISTRA:

   – vhodni kvantni register transformiramo z inverzno kvantno Fourierovo transformacijo, ki tvori superpozicijo vseh možnih števil v vhodnem registru.

   – **DEFINICIJA:** Kvantna Fourierova transformacija splošno superpozicijo $\sum_{x=0}^{Q} \alpha_x |x\rangle$ vhodnega registra pretvori v novo superpozicijo $\frac{1}{\sqrt{Q}} \sum_{z=0}^{Q} \sum_{x=0}^{Q} \alpha_x e^{\frac{i2\pi zx}{Q}} |z\rangle$, torej

$$\sum_{x=0}^{Q} \alpha_x |x\rangle \underset{QFT}{\rightarrow} \frac{1}{\sqrt{Q}} \sum_{z=0}^{Q} \sum_{x=0}^{Q} \alpha_x e^{\frac{i2\pi zx}{Q}} |z\rangle$$

   – po tej operaciji je v našem primeru vhodni kvantni register torej v stanju

$$\frac{1}{Q} \sum_{z} \sum_{b} e^{\frac{i2\pi z(x_0+b\cdot r)}{Q}} |z\rangle$$

   saj so bila prej v vhodnem registru samo števila $x = x_0 + rb$ (vsa ostala so imela amplitudo verjetnosti $\alpha_x = 0$).

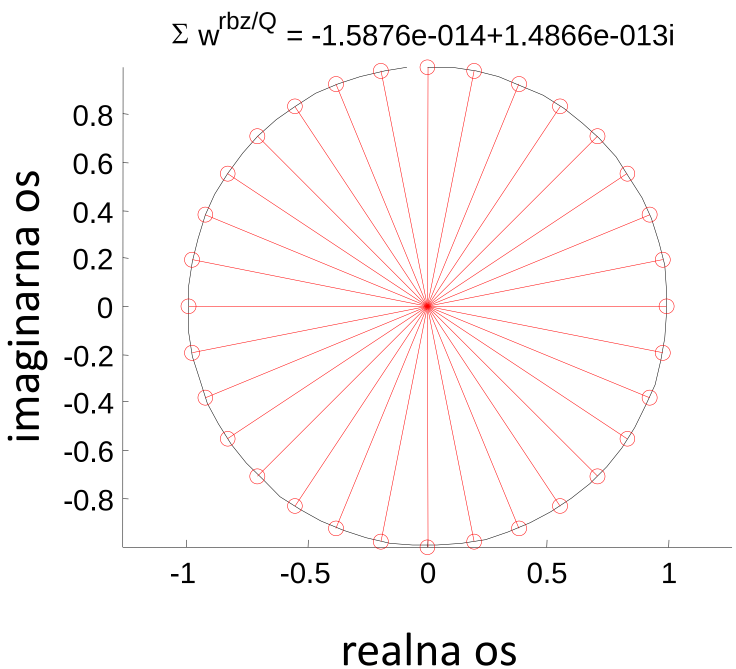
   – izhodni register še vedno vsebuje eno samo vrednost $y_0 = f(x_0)$

Shor-ov algoritem: Kvantni del

6. MERITEV VHODNEGA REGISTRA:
   – izmerimo vhodni register. Velja

$$\frac{1}{Q}\sum_z\sum_b e^{\frac{i2\pi z(x_0+b\cdot r)}{Q}}|z\rangle=\frac{1}{Q}\sum_z e^{\frac{i2\pi z x_0}{Q}}\sum_b e^{\frac{i2\pi z b r}{Q}}|z\rangle$$

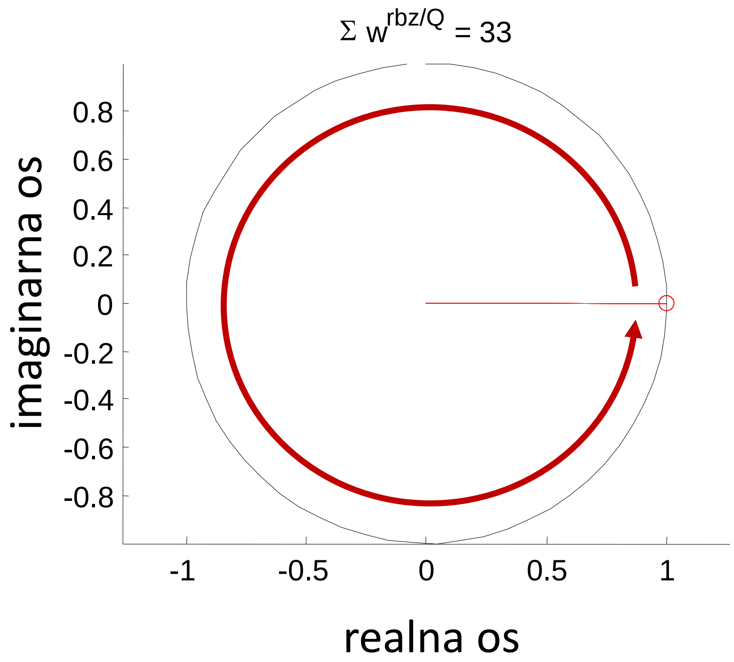
amplitude verjetnosti vseh tistih števil z, za katere velja, da $\frac{zr}{Q}$ ni blizu pozitivnemu celemu številu, bodo v vsoti preko b-ja tvorile 2D enotske vektorje vseh možnih orientacij:



zato se bodo v vsoti preko b-ja izničile in bo njihova vsota enaka ali vsaj blizu 0 (zaradi končnosti vsote, ki izvira iz končnosti vhodnega kvantnega registra ni rečeno, da bo čisto enaka 0).

Shor-ov algoritem: Kvantni del

Amplitude verjetnosti vseh tistih števil z, za katere velja, da je $\frac{zr}{Q}$ zelo blizu pozitivnemu celemu številu (idealno $\frac{zr}{Q}$ = celo število c), pa bodo v vsoti preko b-ja tvorila konstruktivno superpozicijo, zato se bo njihova verjetnost precej ojačala:



Torej je veliko verjetneje, da bomo ob meritvi v vhodnem registru izmerili takšno število z_0, da bo veljalo $\frac{z_0 r}{Q} = c$, kjer je c celo število.

Shor-ov algoritem: Kvantni del

7. <u>OCENITEV PERIODE r:</u>

   – Z veliko verjetnostjo torej velja $\frac{z_0}{Q} = \frac{c}{r}$ in ker mora biti perioda r manjša od N, velja tudi $r<N$. Pri tem sta c in r celi števili.

   – s pomočjo verižnih ulomkov najdemo takšen približek $\frac{c}{r} \approx \frac{z_0}{Q}$, da velja $r<N$. Običajno dobimo več kandidatov za r in preveriti moramo, kateri med njimi izpolnjuje pogoj $f(x) = f(x+r)$.

   – če nismo uspešni ponovimo celoten kvantni del Shorovega algoritma

Aritmetika po modulu *N* & Kvantna vezja

- V kvantnem registru velikosti *N* je vsota po modulu 2^N ena izmed najbolj splošnih unitarnih operacij (xor je vsota po modulu 2):

$$x \in \{0, 1\}^n \quad \text{and} \quad a \in \{0, 1\}^n$$

$$|x\rangle \rightarrow |\,(x + a) \bmod 2^n\rangle$$

- Shor je uporabil algoritem zaporednega kvadriranja za implementacijo funkcije $f(x)=a^x \bmod N$
- Implementacija kvantnega vezja za funkcijo *f(x)* je precej bolj kompleksna od DFT in zahteva tudi več kvantnih vrat (specifično vezje za vsako izbrano osnovo *a*)

Shor-ov algoritem: nekaj lastnosti

1. Shor-ov algoritem je nedeterminističen (*probabilistic*). Ne najde vedno netrivialnega faktorja števila N (trivialna faktorja števila 21, na primer, sta 1 in 21, 7 in 3 pa sta netrivialna faktorja).

   Na primer, fakorizirajmo število 15 z x = 14. Potem se bodo v izhodnem registru vrstila naslednja zaporedja funkcije f(x):

   1, 14, 1, 14, 1, 14, …

   Perioda r = 2, torej sta edina faktorja števila 15, ki jih vrne Shorov algoritem gcd(14-1, 15) = 1, in gcd(14+1, 15) = 15, torej trivialna faktorja števila 15.

2. Kvantno vezje Shorovega algoritma je specifično za vsak *N* in naključno vrednost *a* v funkciji $f(x) = a^x \bmod N$

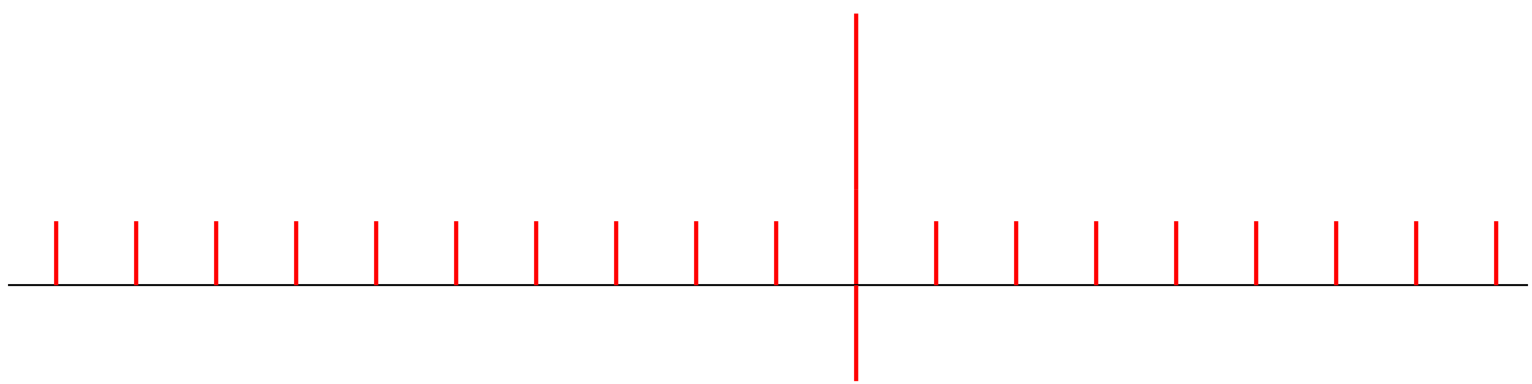
3. Časovna zahtevnost Shorovega algoritma je $O((\log N)^3)$

4. Peter Shor je leta 1999 za svoj algoritem in njegov pridonos k teoretičnemu računalništvu prejel **Gödelovo nagrado**.

Grover-jev kvantni algoritem

Kvantno iskanje po podatkovni bazi.

Najde element v podatkovni bazi v $O(\sqrt{n})$ poizvedbah.



Kakršenkoli klasičen algoritem, determinističen ali ne, potrebuje v povprečju $O(n)$ poizvedb!

Grover-jev kvantni algoritem

1. Postavi kvantni register v stanje superpozicije vseh indeksov:

$$|\omega\rangle = \frac{1}{\sqrt{Q}} \sum_{x=0}^{Q-1} 1\,|x\rangle$$

2. S pomočjo označevalne funkcije $f(x)$ spremenimo predznak amplitude verjetnosti indeksa iskanega elementa. Predznake amplitud verjetnosti indeksov ostalih elementov pustimo nespremenjene.

$$\alpha_x|x\rangle \rightarrow -\alpha_x|x\rangle, \qquad \text{če} \quad f(x) = 1\,,$$

$$\alpha_x|x\rangle \rightarrow \alpha_x|x\rangle\,, \qquad \text{če} \quad f(x) = 0\,.$$

3. Izračunamo inverz amplitud verjetnosti vseh indeksov okoli njihove povprečne vrednosti $\bar{\alpha}$:

$$\forall x\colon \;\; \alpha_x = 2 \cdot \bar{\alpha} - \alpha_x \qquad\qquad \bar{\alpha} = \frac{1}{Q} \sum_{x=0}^{Q-1} \alpha_x$$

Koraka 2 in 3 ponovimo $\frac{\pi}{4}\sqrt{\frac{Q}{k}}$–krat, kjer je k število elementov v bazi, ki so enaki iskanemu elementu. Omenjeno število iteracij je dokazano optimalno in ga ni priporočljivo preseči.

Grover-jev kvantni algoritem - Zgled 1

Zgled: Dana je baza šestnajstih skritih gesel. V njej želimo poiskati geslo, ki dešifrira niz zakodiranih znakov. Elementom baze dodelimo indekse od 0 do 15. Predpostavimo, da naš zakodirani niz znakov dešifrira samo geslo, ki je v bazi shranjeno v elementu z indeksom 4. Imamo torej naslednjo označevalno funkcijo:

$$f(x) = \begin{cases} 1, & ko\ x = 4 \\ 0, & drugače \end{cases}.$$

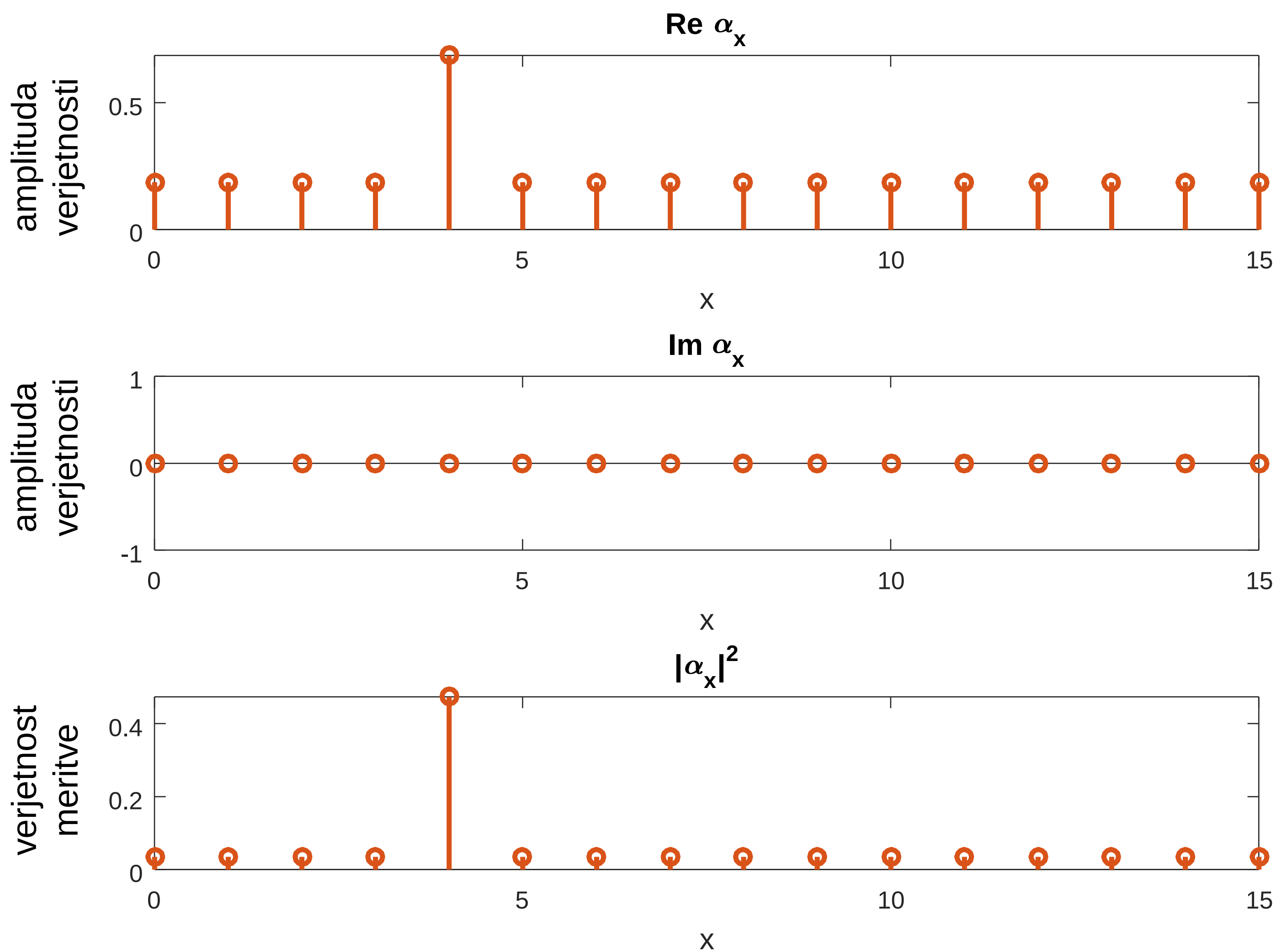
Indekse elementov shranimo v kvantni register z *N*=4 biti. V prvem koraku Groverjevega algoritma postavimo kvantni register v naslednjo superpozicijo stanj:

$$|\omega\rangle = \frac{1}{\sqrt{16}} \sum_{x=0}^{15} 1\,|x\rangle$$

Nato iterativno izvajamo drugi in tretji korak Groverjevega algoritma.

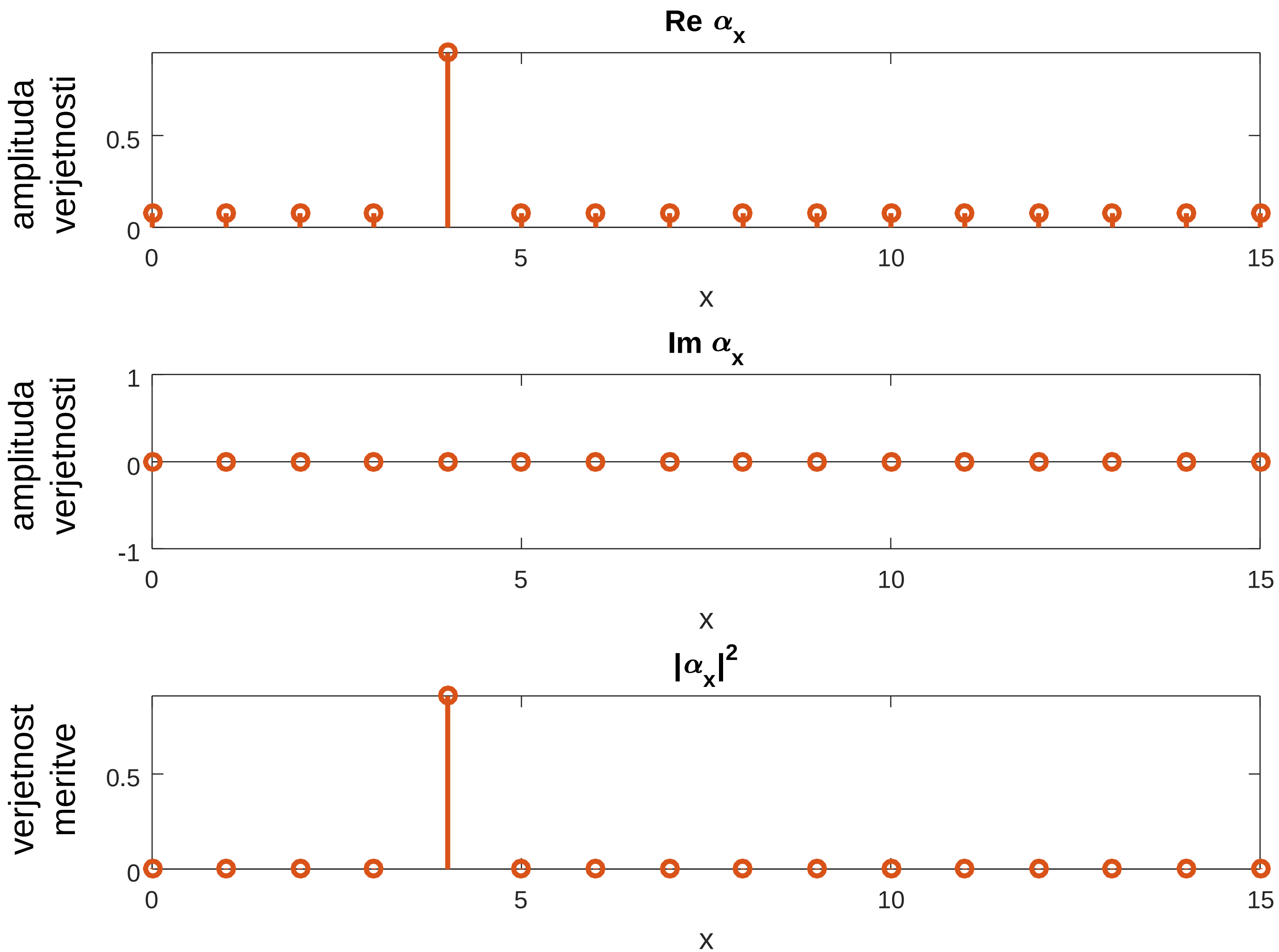
Ker je $\frac{\pi}{4}\sqrt{(\frac{Q}{k})} = \frac{\pi}{4}\sqrt{(\frac{16}{1})} = \pi = 3{,}14$, po tretji iteraciji opravimo meritev.

Grover-jev kvantni algoritem - Zgled 1



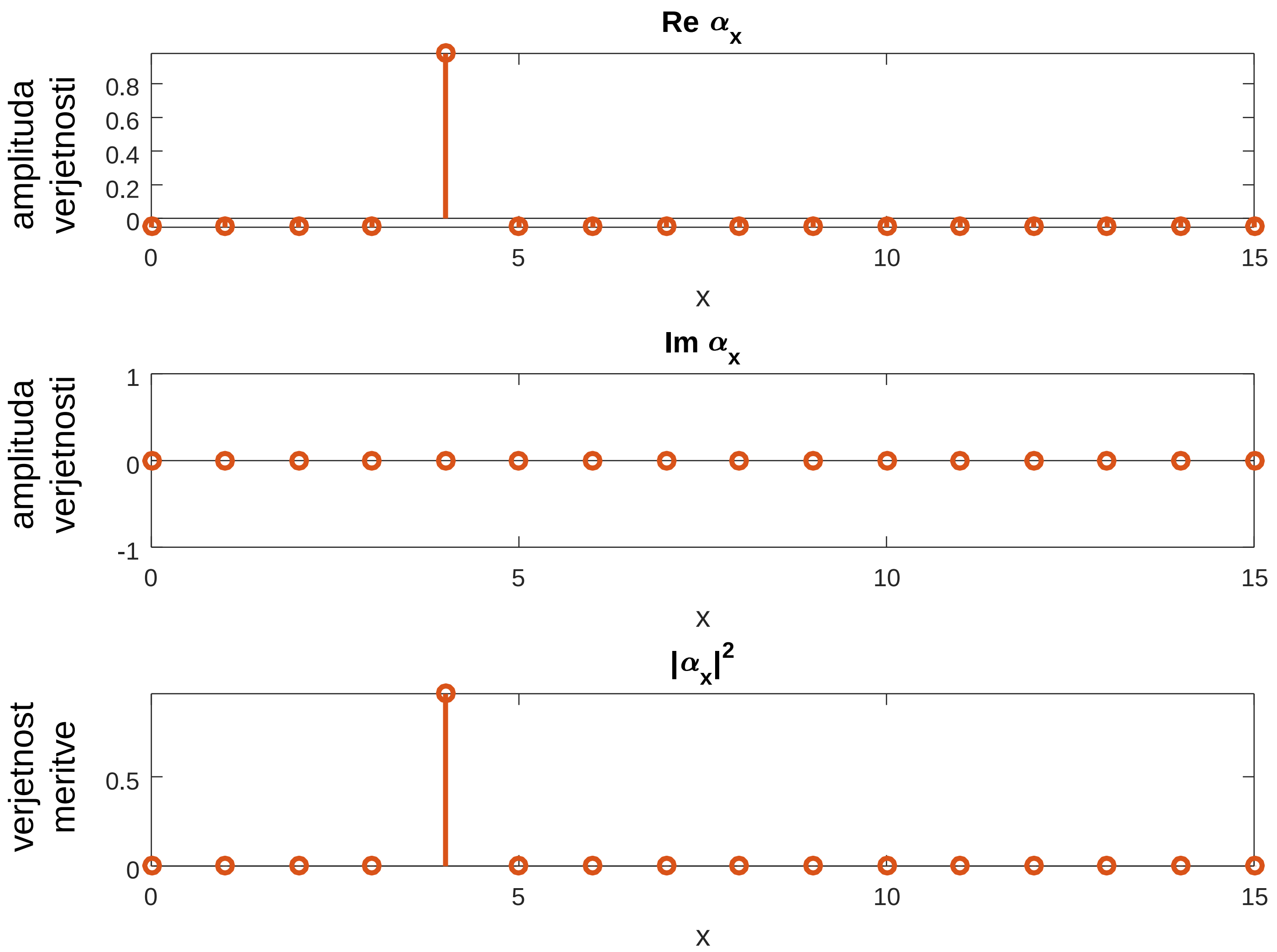
Slika 1: Amplitude verjetnosti in verjetnost meritve posameznega indeksa elementa v bazi po prvi iteraciji drugega in tretjega koraka Groverjevega algoritma.

Grover-jev kvantni algoritem - Zgled 1



Slika 2: Amplitude verjetnosti in verjetnost meritve posameznega indeksa elementa v bazi po drugi iteraciji drugega in tretjega koraka Groverjevega algoritma.

Grover-jev kvantni algoritem - Zgled 1



Slika 3: Amplitude verjetnosti in verjetnost meritve posameznega indeksa elementa v bazi po tretji iteraciji drugega in tretjega koraka Groverjevega algoritma.

Razredi kvantne računske kompleksnosti

- **BQP** (Bounded-Error Quantum Polynomial-Time) je razred odločitvenih problemov rešljivih v polinomskem času na kvantnem računalniku, pri čemer je verjetnost napake manjša ali enaka 1/3.

- Analogno z razredom BPP (“bounded error probabilistic polinimial time“) je izbira mejne verjetnosti 1/3 samo stvar dogovora. Algoritem lahko izvedemo poljubno mnogokrat in izberemo najpogostejši odgovor. Na ta način se lahko verjetnost pravilnega odgovora dvignemo poljubno blizu 1 (**Chernoff-ova zgornja meja** - **Chernoff bound**).

Razredi računske kompleksnosti