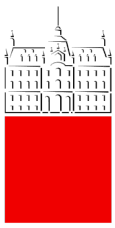
Univerza *v Ljubljani*
Fakulteta za *matematiko in fiziko*



Oddelek za fiziko

Seminar - 3. letnik, I. stopnja

Kvantni računalniki

Avtor: Tomaž Čegovnik

Mentor: prof. dr. Anton Ramšak

Ljubljana, marec 2012

Povzetek

Kvantno računalništvo je zelo mlado področje, saj se je začelo dobro razvijati šele v devetdesetih letih prejšnjega stoletja. Tema je zelo zanimiva, saj kvantni računalniki pri nekaterih problemih močno prekašajo časovno zahtevnost klasičnih računalnikov. Področje je sicer zelo obsežno, zato bom v tem seminarju predstavil samo tistih nekaj osnovnih konceptov, na katerih je zgrajena vsa teorija. Bolj se bom posvetil teoriji, saj se eksperimentalno področje razvija zelo počasi in nepredvidljivo. Na začetku bom predstavil idejo kvantnega bita, nato pa se bom poglobil tudi v enega najodmevnejših algoritmov, s pomočjo katerega bi lahko faktoriziral tudi zelo velika števila.

1 Uvod

Kaj sploh je kvantni računalnik? Najprej lahko seveda rečemo, da je to računalnik, katerega operacije potekajo z zakoni kvantne mehanike. Vendar tudi klasični računalnik deluje pod zakoni kvantne mehanike, pa ga zato ne imenujemo kvantni računalnik. Kvantni računalnik je tisti, ki izkorišča zelo specifične transformacije svojih notranjih stanj, ki jih omogoča kvantna mehanika pod zelo strogimi pogoji.

Kvantni računalniki so uporabni, ker lahko določene računske operacije opravljajo mnogo hitreje kot klasični računalniki. Eno izmed področij, kjer so kvantni računalniki zelo učinkoviti, je faktorizacija velikih števil. To predstavlja določeno nevarnost današnji varni komunikaciji, saj le-ta temelji na algoritmu, ki uporablja velika števila, ki se jih ne da zlahka faktorizirati. Naslednja prednost kvantnih računalnikov je ta, da lahko zelo hitro iščejo po neurejenih bazah. To si lahko predstavljamo kot iskanje v telefonskem imeniku po številkah. Prednost se kaže tudi pri simulaciji kemičnih reakcij. Relativno majhen kvantni računalnik bi simuliral kemične reakcije mnogo hitreje kot najboljši današnji superračunalniki.

2 Fizikalni pogoji

Na tisti sistem, ki pri kvantnem računalniku predstavlja logični bit, ne sme delovati nobena fizikalna interakcija, ki ni pod popolnim nadzorom programa. Vse interakcije, ki so pri navadnem računalniku popolnoma irelevantne, tukaj prinašajo katastrofalne motnje v potek operacij kvantnega računalnika. To so lahko že molekule zraka, ki se odbijajo od fizičnih sistemov - le-ti predstavljajo kvantne logične bite - ali pa absorpcija termične energije. Vse to pelje do *dekoherence*, ki je usodna za kvantno računalništvo.

Da se izognemo omenjenim težavam, je potrebno naše fizikalne sisteme, ki predstavljajo logične bite, ustrezno izolirati od vseh motenj. Izolacijo lahko dosežemo tako, da uporabimo sistem na atomskem nivoju z majhnim številom stanj. Najbolj uporabni so dvonivojski sitemi, saj lahko take sisteme popolnoma nadzorujemo in so veliko bolj neobčutljivi na motnje.

Predvsem pa sta tukaj pomembni dve stvari. Prva je ta, da je razlika med energijskimi nivoji na atomski skali mnogo večja kot pri večjih sistemih. Drugi razlog pa je, da lahko šibke zunanje motnje popravljamo. Pri klasičnem računalništvu je korekcija napak nekaj normalnega, pri kvantnem računalniku pa je toliko bolj težavna, ker ne poznamo niti originalnega, niti poškodovanega stanja.

3 Kubit

Kvantni bit - kubit (angl. Qbit oz. Qubit, odvisno od literature) se od klasičnega bita razlikuje v tem, da nima samo dveh diskretnih vrednosti $|0\rangle$ in $|1\rangle$, ampak je lahko tudi superpozicija teh dveh vrednosti. To na splošno opišemo s stanjem

$$|\psi\rangle = \alpha_0|0\rangle + \alpha_1|1\rangle = \begin{bmatrix}\alpha_0\\ \alpha_1\end{bmatrix}, \tag{3.1}$$

kjer sta α_0 in α_1 kompleksni števili, za kateri velja normalizacija

$$|\alpha_0|^2 + |\alpha_1|^2 = 1. \tag{3.2}$$

Stanje $|\psi\rangle$ je *superpozicija* stanj $|0\rangle$ in $|1\rangle$ z amplitudama α_0 in α_1. Če je eden od α_0 ali α_1 enak nič, dobimo kubit ki ima eno od stanj klasičnega bita.

Stanje je lahko kakršnakoli superpozicija ortogonalnih stanj, to pomeni za dva kubita

$$|\Psi\rangle = \alpha_{00}|00\rangle + \alpha_{01}|01\rangle + \alpha_{10}|10\rangle + \alpha_{11}|11\rangle = \begin{bmatrix} \alpha_{00} \\ \alpha_{01} \\ \alpha_{10} \\ \alpha_{11} \end{bmatrix}, \tag{3.3}$$

kjer so amplitude med sabo povezane z normalizacijo

$$|\alpha_{00}|^2 + |\alpha_{01}|^2 + |\alpha_{10}|^2 + |\alpha_{11}|^2 = 1. \tag{3.4}$$

Splošno lahko potem za n kubitov rečemo, da je to superpozicija 2^n različnih stanj, kjer so amplitude normirane:

$$|\Psi\rangle = \sum_{x=1}^{2^n} \alpha_x |x\rangle_n, \tag{3.5}$$

$$\sum_{x=1}^{2^n} |\alpha_x|^2 = 1 \tag{3.6}$$

To pa je le ena izmed oblik splošnega stanja. Lahko se tudi zgodi, da sta kubita kvantno prepletena. To pomeni, da ju kljub njuni prostorski ločenosti, ne moremo obravnavati posamično, ampak v skupnem stanju.

Če imamo dva kubita, enega v stanju $|\psi\rangle = \alpha_0|0\rangle + \alpha_1|1\rangle$ in drugega v stanju $|\phi\rangle = \beta_0|0\rangle + \beta_1|1\rangle$, potem je tenzorski produkt teh dveh stanj:

$$\begin{aligned} |\Psi\rangle &= |\psi\rangle \otimes |\phi\rangle = (\alpha_0|0\rangle + \alpha_1|1\rangle) \otimes (\beta_0|0\rangle + \beta_1|1\rangle) \\ &= \alpha_0\beta_0|00\rangle + \alpha_0\beta_1|01\rangle + \alpha_1\beta_0|10\rangle + \alpha_1\beta_1|11\rangle \\ &= \begin{bmatrix} \alpha_0\beta_0 \\ \alpha_0\beta_1 \\ \alpha_1\beta_0 \\ \alpha_1\beta_1 \end{bmatrix} \end{aligned} \tag{3.7}$$

Pojavu kvantne prepletenosti se želimo pri našem modelu kvantnega računalnika izogniti, saj se lahko začnejo kubiti obnašati zelo čudno.

Klasični bit oz. par bitov pa je lahko samo v enem izmed stanj $|00\rangle, |01\rangle, |10\rangle, |11\rangle$.

Tukaj se že kaže čar kvantnih bitov, saj so lahko v dveh kubitih naenkrat vsa štiri stanja hkrati, medtem ko lahko klasična bita zasedeta eno samo stanje.

3.1 Reverzibilne operacije

Kvantni računalniki velik del svoje čarovnije naredijo skozi reverzibilne operacije. Obstaja samo en ireverzibilen proces, ki se imenuje *meritev*. Meritev je edini postopek, ki nam da informacijo o stanju kubita. Več o mertivi sem opisal v posebnem poglavju.

Reverzibilne operacije pri klasičnem bitu so:

- negacija $\mathbf{X}$:$|1\rangle \rightarrow |0\rangle$;
- identiteta
- permutacije $\mathbf{S}_{10}|xy\rangle = |yx\rangle$

Reverzibilne operacije na kubitih so vse možne unitarne transformacije:

$$\mathbf{U}\mathbf{U}^\dagger = \mathbf{U}^\dagger\mathbf{U} = 1. \tag{3.8}$$

Vse reverzibilne operacije na klasičnih bitih lahko povežemo z unitarnimi operacijami na kubitih. Transformacije NOT (negacija), SWAP (zamenjava) in cNOT (kontrolirana negacija) lahko enako definiramo tako za kubite, kot tudi za klasične bite. Enako velja tudi za Hadamardovo transformacijo, ki je sestavljena iz NOT in Z operatorja. Vse omenjene operatorje v kvantnih algoritmih imenujemo kvantna logična vrata. Podrobno so opisana v naslednjem poglavju.

4 Kvantna logična vrata

4.1 Hadamardova vrata

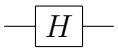
$$H = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \tag{4.1}$$

Deluje na stanja $|0\rangle$ in $|1\rangle$ na naslednji način:

$$H|0\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}} \tag{4.2}$$

$$H|1\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}} \tag{4.3}$$

V kvantnem diagramu imajo oznako



4.2 SWAP vrata

SWAP vrata zamenjajo stanja dveh kubitov.

$$\mathbf{S}_{10} = \mathbf{S}_{01} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.4}$$

4.3 cNOT vrata

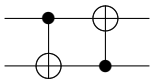
CNOT (controlled NOT) vrata, delujejo na vsaj dva kubita. En kubit predstavlja kontrolo za drugi kubit tako, da prvi kubit deluje na drugega z NOT operacijo, vendar samo v primeru, ko je prvi kubit v stanju $|1\rangle$. Predstavljena so z matriko:

$$\mathbf{C}_{01} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{4.5}$$

Tako delujejo cNOT vrata na dva kubita:

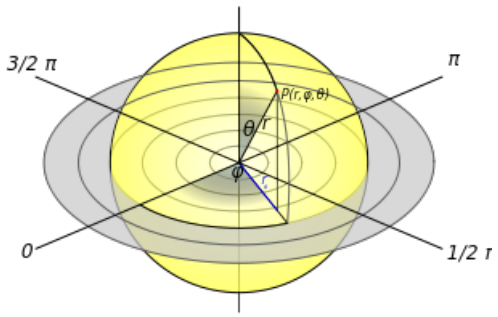
$$\begin{aligned} |00\rangle &\rightarrow |00\rangle \\ |01\rangle &\rightarrow |01\rangle \\ |10\rangle &\rightarrow |11\rangle \\ |11\rangle &\rightarrow |10\rangle \end{aligned} \tag{4.6}$$

V kvantnem diagramu izgledajo cNOT vrata tako:



4.4 Blochova sfera

Da bomo razumeli kako delujejo nekatera vrata moramo najprej vpeljati pojem Blochova sfera. Uporablja se jo v kvantni mehaniki za opis stanja dvonivojskih sistemov. Točka na sferi nam predstavlja stanje našega dvonivojskega sistema. Severni pol pred-



Slika 1: Blochova sfera

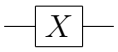
stavlja $|1\rangle$ južni pol pa $|0\rangle$. Točka na ekvatorju predstavlja superpozicijo stanj $|0\rangle$ in $|1\rangle$, kjer imata stanji enako amplitudo. Stanje splošno opišemo z $|\Psi\rangle = sin(\theta/2)e^{-i\phi/2}|0\rangle + cos(\theta/2)e^{i\phi/2}|1\rangle$. Z Blochovo sfero si sicer najbolje predstavljamo usmerjenost spina, vendar lahko z njo opišemo katerikoli drugi dvonivojski sistem.

4.5 Paulijeva X vrata

Paulijeva X vrata predstavljajo vrtenje na Blochovi sferi okoli osi x za π in so ekvivalentna NOT vratom. Preslika $|0\rangle$ v $|1\rangle$ in $|1\rangle$ v $|0\rangle$.

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \tag{4.7}$$

V kvantnem diagramu imajo oznako

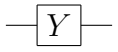


4.6 Paulijeva Y vrata

Paulijeva Y vrata so ekvivalentna vrtenju na Blochovi sferi okrog osi y za π radianov. Preslika $|0\rangle$ v $i|1\rangle$ in $|1\rangle$ v $-i|0\rangle$. Predstavimo jih z matriko:

$$Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} \tag{4.8}$$

V kvantnem diagramu imajo oznako

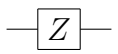


4.7 Paulijeva Z vrata

Paulijeva Z vrata so ekvivalentna vrtenju na Blochovi sferi okrog osi z za π radianov. To je specifičen primer faznih vrat, ko je $\theta = \pi$. $|0\rangle$ ostane nespremenjeno, $|1\rangle$ postane $-|1\rangle$. Matrika je:

$$Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \tag{4.9}$$

V kvantnem diagramu imajo oznako



4.8 Fazna vrata

Fazna vrata ne spremenijo stanja $|0\rangle$, delujejo le na $|1\rangle$ in ga spremenijo v $|e^{i\phi}|1\rangle$. Verjetnost, da izmerimo $|0\rangle$ ali $|1\rangle$, ostane nespremenjena. Fazna vrata spremenijo le fazo stanja. Predstavimo jih z matriko:

$$R_\theta = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\theta} \end{bmatrix}, \tag{4.10}$$

kjer je θ fazni premik. Kot je navedeno že zgoraj, če je $\theta = \phi$, so to Paulijeva Z vrata. V kvantnem diagramu imajo oznako

5 Meritev

Klasični bit ima lahko samo dve stanji: $|0\rangle$ in $|1\rangle$. Da pa opišemo stanje kubita (3.1) potrebujemo mnogo bitov informacij, saj stanje določata dve kompleksni števili α_0 in α_1, ki morata biti normirani (3.2). Očitno je, da imajo kubiti bogatejša stanja od klasičnih bitov, in prav to je razlog, da je kvantni računalnik močnejši od klasičnega računalnika. Ampak pri tem moramo biti previdni!

Če imamo klasične bite, ki predstavljajo 0 ali 1, lahko ugotovimo njihovo stanje tako, da jih pogledamo. V tem ni nič problematičnega. Pridobivanje podatkov iz klasičnega bita na noben način ne more vplivati na njegovo stanje. V katerikoli fazi računanja lahko pridobimo stanje klasičnega bita vendar ne bomo zmotili poteka računanja.

Tukaj pa se kubiti popolnoma razlikujejo od klasičnih bitov. Če imamo n kubitov v superpoziciji (3.3), ne obstaja noben način, da bi izmerili ogromno količino informacij, vsebovanih v amplitudah α_x. Ker ne moremo prebrati vrednosti teh amplitud, ne moremo določiti v kakšnem stanju so kubiti. Edini način, da dobimo kakršnokoli informacijo je, da naredimo *meritev*.

Pri meritvi dobimo za vsak kubit le 0 ali 1. Ta zbirka ničel in enic nam ne pove v kakšen stanju $|\Psi\rangle$ so bili kubiti, saj nam stanje določa le *verjetnost* da bomo dobili enega od rezultatov. Recimo, da imamo štiri kubite; potem dobimo npr. $|1001\rangle$, ne izvemo pa ničesar o amplitudah α_x. Bolj splošno, če imamo stanje n kubitov

$$|\Psi\rangle = \sum_{0 \leq x < 2^n} \alpha_x |x\rangle_n, \tag{5.1}$$

potem je verjetnost, da izmerimo binarno zaporedje x, sledeča:

$$p(x) = |\alpha_x|^2 \tag{5.2}$$

To je osnovno pravilo o pridobivanju podatkov iz kubita, imenovano tudi *Bornovo pravilo*.

Ko pa enkrat opravimo meritev in dobimo rezultat, npr. 1001, potem naši štirje kubiti niso več v stanju $|\Psi\rangle$ ampak v $|1001\rangle$. Po meritvi je osnovno stanje izgubljeno. Taki spremembi valovne funkcije pravimo tudi *kolaps* valovne funkcije.

6 Splošni računski proces

Primerno programiran kvantni računalnik naj bi deloval na število x in tvoril novo število $f(x)$ za določeno funkcijo f. Če je x $n-$bitno celo število in $f(x)$ $m-$bitno celo število, potem potrebujemo vsaj $n + m$ kubitov. Nabor $n-$kubitov imenjuemo *vhodni register*, ki nam predstavlja število x; nabor $m-$kubitov pa imenujemo *izhodni register*, ki nam predstavlja število $f(x)$. Seveda bi pričakovali, da imamo lahko samo en register, na katerem opravljamo računske operacije, venda je dvoregistrska arhitektura mnogo bolj primerna, saj mora kvantni računalnik, razen meritve, opravljati samo reverzibilne operacije, zato se izkaže, da je bolje imeti posebej vhodni in izhodni register s katerima upravljamo istočasno.

Za dejanski računski proces potrebujemo poleg omenjenih $n + m$, veliko dodatnih kubitov, ki jih lahko v tem koraku zanemarimo in pogledamo kaj se zgodi, ko vršimo

transformacijo $\mathbf{U}_f$ na vhodni in izhodni register. Transformacija $\mathbf{U}_f$ deluje na našo bazo $|x\rangle_n|y\rangle_m$ $n+m$ kubitov na naslednji način:

$$\mathbf{U}_f(|x\rangle_n|y\rangle_m) = |x\rangle_n|y \oplus f(x)\rangle_m, \tag{6.1}$$

kjer pomeni $\oplus$ seštevanje po modulu 2 po posameznem bitu oz. XOR, brez prenosa. Bolj natančno povedano - če sta x in y $m-$bitni celi števili katerih j-ta bita sta x_j in y_j, potem je $x \oplus y$ $m-$bitno celo število, katerega $j-$ti bit je $x_j \oplus y_j$. Primer: $1101 \oplus 0111 = 1010$.

Če je začetna vrednost izhodnega registra $y = 0$ potem imamo

$$\mathbf{U}_f(|x\rangle_n|0\rangle_m) = |x\rangle_n|f(x)\rangle_m \tag{6.2}$$

in imamo v izhodnem registru kar funkcijo $f(x)$. Ne glede na y, vhodni register ostane v istem stanju $|x\rangle_n$.

Če sedaj delujemo z Hadamardovo transformacijo (4.1) na stanje z dvema kubitoma, dobimo

$$\begin{aligned}(\mathbf{H}\otimes\mathbf{H})(|0\rangle\otimes|0\rangle) &= (\mathbf{H}|0\rangle)(\mathbf{H}|0\rangle)\\ &= \tfrac{1}{\sqrt{2}}(|0\rangle+|1\rangle)\tfrac{1}{\sqrt{2}}(|0\rangle+|1\rangle)\\ &= \tfrac{1}{2}(|00\rangle+|01\rangle+|10\rangle+|11\rangle)\end{aligned} \tag{6.3}$$

Tako lahko splošno zaspišemo $n-$kratni tenzorski produkt n Hadamardov, ki delujejo na stanje z n kubiti $|0\rangle_n$:

$$\mathbf{H}^{\otimes n}|0\rangle_n = \frac{1}{2^{n/2}}\sum_{0\le x<2^n}|x\rangle_n, \tag{6.4}$$

kjer je

$$\mathbf{H}^{\otimes n} = \mathbf{H}\otimes\mathbf{H}\otimes\cdots\otimes\mathbf{H}. \tag{6.5}$$

Če imamo vhodni register v stanju $|0\rangle_n$ in na njega delujemo z $n-$kratno Hadamardovo transformacijo, dobimo superpozicijo vseh možnih vhodnih vrednosti z enakomerno verjetnostno utežjo $\frac{1}{\sqrt{2^n}}$. Če potem na to superpozicijo delujemo z $\mathbf{U}_f$, kjer je izhodni register poln ničel, dobimo zaradi linearnosti iz enačb (6.2) in (6.4):

$$\begin{aligned}\mathbf{U}_f(\mathbf{H}^{\otimes n}\otimes\mathbf{1}_m)(|0\rangle_n|0\rangle_m) &= \frac{1}{2^{n/2}}\sum_{0\le x<2^n}\mathbf{U}_f(|x\rangle_n|0\rangle_m)\\ &= \frac{1}{2^{n/2}}\sum_{0\le x<2^n}|x\rangle_n|f(x)\rangle_m\end{aligned} \tag{6.6}$$

Ta enačba razlaga pomemben del čarobnosti kvantnega računalništva. Če delujemo na vse kubite, prvotno v stanju $|0\rangle_n$, s Hadamardovimi transformacijami, preden delujemo s tranformacijo $\mathbf{U}_f$ na vsak kubit zase, je rezultat stanje, ki vsebuje 2^n izračunov funkcije f. Če imamo sto kubitov v vhodnem registru v stanju $|0\rangle_{100}$ in sto Hadamardovih logičinih vrat deluje na vhodni register pred izvršitvijo $\mathbf{U}_f$, potem izhodni register vsebuje rezultate $2^{100} \approx 10^{30}$ izračunov funkcije f. Temu rečemo *kvantni paralelizem.*

Ampak velik del tega čudeža kvantnih računalnikov je samo navidezen. Ne moremo trditi, da *je* rezultat kalkulacije 2^n izračunov funkcije f. Rečemo lahko samo, da ti

izračuni karakterizirajo obliko stanja izhodnega registra. Vedeli bi kakšno je stanje, ne bi pa poznali vseh $2n$ izračunov funkcije f. Spomnimo se poglavja o meritvi, da ni nobenega načina, da bi ugotovili v kakšnem stanju je izhodni register. Edini način, da karkoli izvemo je meritev.

Če pošljemo vseh $n+m$ kubitov skozi merilna vrata, nam Bornovo pravilo pove, da bomo bomo izmerili katerokoli vrednost x manjšo od 2^n, v primeru da ima stanje registrov obliko (6.6). Hkrati bo rezultat meritve izhodnega registra vrednost funkcije f za tisti določen x. Tako pri meritvi izvemo eno določeno vrednost funkcije f pri popolnoma naključnem x_0. Po meritvi se stanje registrov reducira v $|x_0\rangle|f(x_0)\rangle$ in ne moremo izvedeti popolnoma ničesar od katerekoli druge vrednosti x.

To oviro bi lahko obšli s kopiranjem registrov in potem opravili več meritev na identičnih stanjih. Tako bi iz rezultatov posameznih meritev skonstruirali celotno stanje z vsemi amplitudami. Vendar nam to kopiranje registrov prepoveduje "no-cloning theorem", ki je dokazan v [1].

7 Kvantni algoritmi

Kvatni algorimi so navadno opisani s kvantnimi vezji oz. diagrami. Kvantni diagram ima nekaj vhodnih kubitov in se ponavadi konča z meritvijo, vsebuje pa tudi osnovna kvantna logična vrata. Kvantne algoritme lahko opišemo tudi z drugimi modeli, vendar so kvantni diagrami najenostavnejši in najpreglednejši model.

Kvantne algoritme lahko razvrstimo glede na tehniko, ki jo uporablja algoritem. Nekaj najbolj pogostih je: kvantna Fourierova transformacija, kvantni sprehodi, amplitudna amplifikacija... Najbolj znana sta Shorov in Groverjev algoritem. Prvega bomo podrobneje opisali kasneje, za Groverjev algoritem pa naj omenim, da je zelo učinkovit pri iskanju v neurejenih podatkovnih bazah.

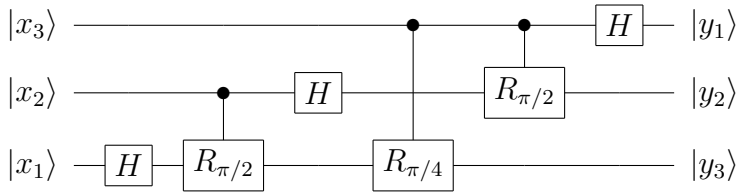
7.1 Kvantna Fourierova transformacija

Najprej si poglejmo kvantno Fourierovo transformacijo, ki je zelo pomembna pri več kvantnih algoritmih, med drugim tudi pri Shorovem algoritmu. Sestavimo jo lahko iz $O(n^2)$ Hadamardovih in faznih vrat, kjer je n število kubitov. Klasično bi potrebovali $O(n2^n)$ vrat, kar je eksponentno več od prejšnje številke. Najboljši algoritmi s fourierovo transformacijo dandanes potrebujejo le $O(n\log n)$ vrat, da dosežejo učinkovit približek.

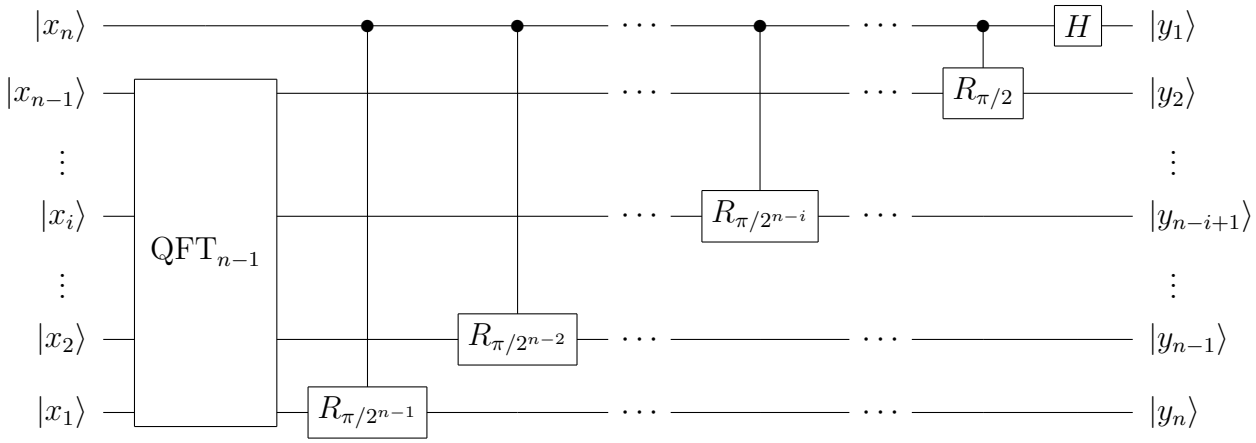
Fourierovo transformacijo definiramo kot unitarno transformacijo podano z

$$\mathbf{U}_{FT}|x\rangle_n = \frac{1}{\sqrt{2^n}} \sum_{y=0}^{2^n-1} e^{2\pi i x y/2^n} |y\rangle_n. \tag{7.1}$$

Primer kvantnega diagrama za 3-kubitno Fourierovo transformacijo.



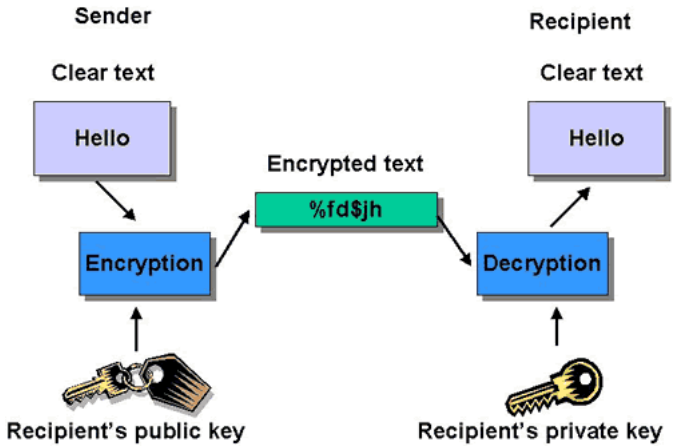
Splošno pa izgleda tako:



7.2 RSA

Na kratko bom opisal RSA kriptografijo, da bomo razumeli, kako nam lahko pomaga Shorov algoritem pri razbijanju tega šifrirnega algoritma.

RSA je algoritem za šifriranje z javnim ključem. Bistvo javnega ključa je, da prejemnik zgenerira javni in zasebni ključ, ki sta povezana. Javnega potem pošlje pošiljatelju, ki z njim širfrira sporočilo. Ko je sporočilo enkrat zaklenjeno z javnim ključem, ga lahko odklenemo le z njemu pripadajočim zasebnim ključem. Nemogoče je odkleniti sporočilo le z javnim ključem.



Slika 2: Javni in zasebni ključ

Postopek kreiranja ključev

1. Izberemo dve različni praštevili, ki sta popolnoma naključni in neodvisni eno od drugega.
2. Izračunamo $n = pq$.
3. Izračunamo koeficient $\Phi(n) = (p-1)(q-1)$.
4. Izberemo celo število e, tako da velja $1 < e < \Phi(n)$, ki je relativno praštevilo številu $\phi(n)$.
5. Izračunamo d, da je $de \equiv 1 \ (\mathrm{mod}\Phi(n))$. To je $de = 1 + k\phi(n)$ za celoštevilčno vrednost k.

Javni ključ je sestavljen iz:

- n, generator
- e, šifrirni eksponent

Zasebni ključ je sestavljen iz:

- n, generator
- d, dešifrirni eksponent

Sporočilo šifriramo s formulo:

$$c = m^e \ \mathrm{mod}n, \tag{7.2}$$

kjer je c šifrirano sporočilo in m čisto sporočilo.
Šifrirano sporočilo dešifriramo s formulo

$$m = c^d \ \mathrm{mod}n \tag{7.3}$$

7.3 Shorov algoritem

Shorov algoritem, imenovan po matematiku Petru Shoru, je algoritem za faktorizacijo celih števil. Problem je torej sledeč. Imamo naravno število N, iščemo taka naravna števila $p_1, p_2, \ldots, p_l, r_1, r_2, \ldots, r_l$, da bojo vsi p_j praštevila in $N = p_1^{r_1} p_2^{r_2} \ldots p_l^{r_l}$.

Najprej s klasičnim algoritmom reduciramo problem na takega, ki ga lahko rešimo s kvantnim algoritmom.

Klasični del algoritma

1. Izberi številko x med $1 < x < n$.
2. Poiščemo največji skupni delitelj od (x, n).
3. Poišči najmanjši r, da $x^r \equiv 1(\mathrm{mod}n)$.
4. Začni znova pri ena, če je r lih ali je $x^{r/2} \equiv -1(\mathrm{mod}n)$.
5. Vrni največji skupni delitelj $(x^{r/2} - 1, n)$.

Kvantni del algoritma

1. Določi q kot potenco 2 s pogojem $n^2 \leq q < 2n^2$.
2. Inicializiraj vhodni register s superpozicijo vseh stanj $a \pmod{q}$. Dobimo stanje:

$$\frac{1}{\sqrt{q}} \sum_{a=0}^{q-1} |a\rangle|0\rangle. \tag{7.4}$$

3. Inicializiramo izhodni register s superpozicijo stanj $x^a \pmod{n}$. Dobimo:

$$\frac{1}{\sqrt{q}} \sum_{a=0}^{q-1} |a\rangle|x^a \pmod{n}\rangle. \tag{7.5}$$

4. Na vhodni register delujemo s kvantno Fourierovo transformacijo, pri čemer dobimo:

$$\frac{1}{q} \sum_{a=0}^{q-1} \sum_{c=0}^{q-1} e^{2\pi iac/q} |c\rangle|x^a \pmod{n}\rangle \tag{7.6}$$

5. Opravimo meritev. Lahko pišemo $a = br + k$. Verjetnost, da dobimo r, je največja po $O(\log \log n)$. Natančen izračun je opisan v [4].
6. Ponavljamo, dokler ne dobimo pravega razultata.

Literatura

[1] N.D. Mermin, *Quantum Computer Science*, Cambridge: Cambridge University Press, 2007.

[2] Michael A. Nielsen in Isaac L. Chuang, *Quantum Computation and Quantum Information*, Cambridge: Cambridge University Press, 2000.

[3] M. Mosca, *Quantum Algorithms*, http://arxiv.org/abs/0808.0369, University of Waterloo and St. Jerome's University, 2008.

[4] P.W. Shore, *Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer*, SIAM Journal on Scientific and Statistical Computing 26: 1484, 1996.