

KVANTNI ALGORITMI

MIHA POMPE

Fakulteta za matematiko in fiziko
Univerza v Ljubljani

Cilj članka je proučiti delovanje kvantnih računalnikov in algoritmov ter se poglobiti v dva računski modela: model kvantnih vrat in adiabatični kvantni računalnik. Z uporabo obeh modelov sta rešena dva problema: Deutsch problem in optimizacija portfelja. Pri obeh primerih je izpeljana teoretična osnova, ki omogoča rešiti problem, ter kako le-to praktično implementiramo v kvantnem računalniku. Na kratko je navedeno tudi, kako so kvantni računalniki narejeni.

QUANTUM ALGORITHMS

The aim of this paper is to examine how quantum computers and algorithms work, as well as to look into two models of computation: quantum gate array and adiabatic quantum computer. The models are used to solve two problems: Deutsch problem and financial portfolio optimization. The solutions for both problems are derived and it is shown how to implement them in a real quantum computer. Two quantum computer implementations are briefly discussed.

1. Uvod

Kvantni računalniki so naprave, ki izkoriščajo lastnosti kvantne fizike za shranjevanje podatkov in izvajanje izračunov. Za reševanje določenih problemov nam te lastnosti zelo pomagajo do te mere, da prekašajo celo najboljše superračunalnike. Klasični računalniki shranjujejo informacijo v bitih, ki so lahko enaki 0 ali 1, medtem ko je v kvantnem računalniku osnovna enota pomnilnika kvantni bit ali kubit.

Klasične bite predstavimo z napetostjo, na primer 0 V predstavlja stanje 0 in 5 V predstavlja stanje 1. Za implementacijo kvantnih bitov uporabimo kvantne lastnosti fizičnih sistemov, kot je spin elektrona ali polarizacija fotona. Ker so to kvantni sistemi, je lahko njihovo stanje superpozicija več lastnih stanj.

Primerjajmo klasični in kvantni računalnik s štirimi biti. V klasičnem računalniku lahko s štirimi biti zapišemo katerokoli celo število med 0 in 15. Štirje kubitovi sestavljajo sistem s 16 lastnimi stanji. Zaradi kvantne narave kubita (superpozicija lastnih stanj) lahko v kvantnem računalniku zapišemo več števil hkrati. Na primer imamo lahko stanje ($|\psi\rangle$), ki je superpozicija stanja $|3\rangle$ in $|8\rangle$, $|\psi\rangle = \frac{1}{\sqrt{2}}|3\rangle + \frac{1}{\sqrt{2}}|8\rangle$.

Prav zaradi te lastnosti imajo kvantni računalniki v določenih situacijah prednost pred klasičnimi. Kadar obstaja veliko kombinacij, lahko kvantni računalnik vse primere obravnava hkrati. Primeri, kjer izkoriščamo to prednost, so faktorizacija velikih števil in iskanje optimalne poti med dvema krajema. Opozoriti je treba, da obstaja veliko problemov, kjer so klasični računalniki bolj uporabni od kvantnih.

Za zdaj imajo kvantni računalniki še kar nekaj težav, kot so občutljivost na toploto, elektromagnetna polja in trke z molekulami v zraku, kar povzroči, da kubit izgubi informacijo o shranjenem stanju. Ta proces imenujemo dekoherenca, ki povzroči kolaps sistema. Kako hitro nastopi dekoherenca, je tudi zelo odvisno od števila kubitov. Zaznava in odpravljanje napak, ki so posledica dekoherence, je problem, s katerim se soočajo vse vrste kvantnih računalnikov.

V tem članku se bomo posvetili kvantnim računalnikom in algoritmom. V prvem delu bomo najprej formulirali zapis in lastnosti kubita, nato pa bomo proučili dva modela kvantnega računanja: model kvantnega vezja in adiabatični kvantni računalnik. Za boljše razumevanje so predstavljeni tudi primeri algoritmov in postopek evalvacije problema.

2. Kubit

Osnovni element kvantnega računalnika je kvantni bit ali kubit (ang. *qubit*). Analogno klasičnemu bitu ima tudi kubit dve stanji, ki sta lastni stanji poljubnega dvonivojskega kvantnega sistema. Primera takšnega sistema sta spin elektrona s stanjema gor in dol ter polarizacija fotona, ki je lahko polariziran vodoravno ali navpično. Kubit ($|\psi\rangle$) v Diracovem zapisu predstavimo kot superpozicijo stanj $|0\rangle$ in $|1\rangle$:

$$|\psi\rangle = c_0 |0\rangle + c_1 |1\rangle,$$

kjer sta $c_0, c_1 \in \mathbb{C}$ in zanju velja normalizacija, $|c_0|^2 + |c_1|^2 = 1$. Zapis lahko poenostavimo, da velja $|0\rangle = [1, 0]^T$ in $|1\rangle = [0, 1]^T$, s čimer zapis kubita prevedemo na $|\psi\rangle = [c_0, c_1]^T$.

Kubit si lahko predstavljamo kot vektor na Blochovi sferi, ki lahko zaseda poljubno stanje. Katerakoli operacija na kubitom pomeni rotacijo vektorja na sferi, ki jo lahko zapišemo z unitarno matriko. Če primerjamo kubit s klasičnim bitom, ki lahko zaseda le dve diskretni stanji, slednji navidezno vsebuje manj informacije. Izkaže se, da oba vsebujeta le en bit informacije, razlikujeta pa se v tem, kako ju manipuliramo, ter posledično, kakšne algoritme izvajamo z njima.

Stanja več kubitov zapišemo kot tenzorski produkt. Za dva kubita, $|\psi\rangle_a = c_{a0} |0\rangle_a + c_{a1} |1\rangle_a$ in $|\phi\rangle_b = c_{b0} |0\rangle_b + c_{b1} |1\rangle_b$, torej velja

$$\begin{aligned} |\zeta\rangle &= |\psi\rangle_a \otimes |\phi\rangle_b = (c_{a0} |0\rangle_a + c_{a1} |1\rangle_a) \otimes (c_{b0} |0\rangle_b + c_{b1} |1\rangle_b) = \\ &= c_{00} |0\rangle_a \otimes |0\rangle_b + c_{01} |0\rangle_a \otimes |1\rangle_b + c_{10} |1\rangle_a \otimes |0\rangle_b + c_{11} |1\rangle_a \otimes |1\rangle_b \\ &= c_{00} |00\rangle + c_{01} |01\rangle + c_{10} |10\rangle + c_{11} |11\rangle. \end{aligned}$$

Tu smo poenostavili zapis, kjer smo na primer stanje $|0\rangle_a \otimes |1\rangle_b$ zapisali kot $|01\rangle$. Enako, kot smo storili z enim kubitom, lahko stanja dveh kubitov predstavimo z vektorjem: $|00\rangle = [1, 0, 0, 0]^T$, $|01\rangle = [0, 1, 0, 0]^T$, $|10\rangle = [0, 0, 1, 0]^T$ in $|11\rangle = [0, 0, 0, 1]^T$. Poljubno stanje dveh kubitov zapišemo kot $|\zeta\rangle = [c_{00}, c_{01}, c_{10}, c_{11}]^T$. Notacijo lahko splošimo na poljubno število bitov.

Poljubnega stanja dveh ali več kubitov v splošnem ne moremo zapisati v produktni bazi kot $|\Psi\rangle_{ab} = |\psi\rangle_a |\phi\rangle_b$. Torej za dane koeficiente c_{00} , c_{01} , c_{10} in c_{11} ne moremo določiti koeficientov c_{a0} , c_{a1} , c_{b0} in c_{b1} . Takšnemu stanju pravimo, da je prepleteno. Primer takšnega stanja je $|\psi\rangle = |01\rangle - |10\rangle$, torej $c_{01} = 1$ in $c_{10} = -1$.

Sistem z n kubitom ima 2^n možnih stanj, torej je stanje le-tega superpozicija 2^n lastnih stanj oziroma ima vektor njegovega stanja 2^n komponent. Ko izmerimo stanje sistema, kolapsiramo superpozicijo in izmerimo le eno od 2^n možnih stanj s točno določeno verjetnostjo. Prav v tem se kubit razlikuje od klasičnega bita, ki zavzame določeno stanje z verjetnostjo 1 [1].

3. Model kvantnih vrat

Obstaja več modelov računanja za kvantne računalnike. Ti se razlikujejo po osnovnih elementih, na katere je izračun razdeljen. Model računanja pove, kako računalnik izračuna izhod določene funkcije glede na dane vhodne podatke. V kvantnem računalništvu poznamo štiri glavne modele, v tem članku pa se bomo omejili na dva najpogostejša: model kvantnih vrat in adiabatni kvantni računalnik.

3.1 Kvantna logična vrata

Kvantna logična vrata so unitarni operatorji, ki jih lahko zapišemo z unitarno matriko. Logična vrata zavrtijo kubit v določeni smeri na Blochovi sferi, delujejo pa lahko na enega ali več bitov, podobno kot klasična logična vrata. V nasprotju z njimi morajo biti vsa kvantna vrata reverzibilna (bijektivna). Edina ireverzibilna operacija je meritev končnega stanja.

Obstaja neskončno mnogo logičnih vrat, ampak pogledajmo le nekaj najpogostejših. Paulijeva vrata (X, Y in Z) predstavimo s Paulijevimi matrikami (σ_x , σ_y in σ_z) in delujejo na en kubit. Vsaka od njih predstavlja rotacijo za π okoli ene od osi x , y , ali z na Blochovi sferi. Vrata X, imenovana tudi vrata NOT, negirajo lastna stanja oziroma jih zavrtijo okoli osi x , $X|0\rangle = |1\rangle$ in $X|1\rangle = |0\rangle$. Druga pogosto uporabljena vrata so Hadamardova vrata (H), ki lastno stanje spravijo v superpozicijo

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, \quad H|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle), \quad H|1\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle).$$

Med eno bitnimi vrati pogosto navajajo tudi fazna vrata (P_ϕ), za katera velja $P_\phi = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\phi} \end{bmatrix}$. Fazna vrata zasukajo kubit v vodoravni ravnini, $P_\phi|0\rangle = |0\rangle$ in $P_\phi|1\rangle = e^{i\phi}|1\rangle$.

Poglejmo si še nekaj pogostih večbitnih vrat. V našem primeru se bomo omejili na vrata, ki delujejo na dva kubita. Prva so vrata CNOT oziroma nadzorovana vrata NOT, ki negira drugi bit le, če je prvi bit v stanju $|1\rangle$. Na primer $CNOT|11\rangle = |10\rangle$ ali $CNOT|01\rangle = |01\rangle$. Z matriko vrata CNOT zapišemo, kot so prikazana v enačbi (1). Ta vrata delujejo enako kot klasična vrata XOR.

$$CNOT = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad SWAP = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

Pogosta so tudi vrata SWAP (prikazana v enačbi (1)), ki zamenjajo stanji obeh kubitov, kar pomeni $c_{10} \leftrightarrow c_{01}$. V nadaljevanju bomo uporabili tudi Hadamardova vrata za več kubitov. Ta pretvorijo lastno stanje v superpozicijo vseh lastnih stanj.

Več kvantnih vrat lahko vežemo zaporedno, da tvorimo vezje ali algoritem. Računsko to naredimo tako, da matrike vrat množimo med seboj. Nastali produkt je ponovno unitarna matrika.

V klasičnem računalniku lahko katerakoli vrata in posledično tudi katerikoli algoritem sestavimo iz izključno NOR (negiran ali) ali NAND (negiran in) vrat. Podobno lahko naredimo tudi na kvantnem računalniku, le da tu univerzalni set vrat sestavljajo CNOT, Hadamardova in fazna vrata [1].

3.2 Primeri uporabe kvantnega računalnika

Superpozicija kvantnega stanja in uporaba kvantnih vrat na celotnem sistemu omogoča reševanje kombinatoričnih problemov. Ti so na klasičnem računalniku zelo težko rešljivi, saj njihova časovna kompleksnost običajno narašča eksponentno s številom parametrov. Na naslednjih primerih bomo videli, kako lahko izkoristimo prednosti kvantnih računalnikov za reševanje zahtevnih problemov.

Kombinatoriko uporabljamo že od nekdaj za kriptografijo. Faktorizacija naravnih števil je eden glavnih konceptov pri prenosu javnih ključev, saj klasični računalniki skoraj ne morejo faktorizirati velikih števil, na primer produkta dveh praštevil s 300 ciframi. Za kvantne računalnike je bil razvit Shorov algoritem za faktorizacijo, ki ima polinomsko časovno kompleksnost [6]. Z njim bi lahko v eksponentno krajšem času dešifrirali sporočila, zašifrirana z RSA ali Diffie-Hellmanovim algoritmom. Zaradi potencialne nevarnosti se je začel razvoj kriptografskih algoritmov, ki so nerešljivi s kvantnim računalnikom.

Razvoj na področju kemije in biologije temelji na odkritju in manipulaciji molekul. Sistematičen pristop k odkrivanju novih spojin zahteva simuliranje gibanja in interakcij med posameznimi delci, in sicer na nivoju kvantne mehanike. Z večanjem števila molekul v simulaciji število možnih kombinacij raste eksponentno. Zaradi velikega števila kombinacij in kvantne narave problema je kvantni

računalnik primeren za reševanje teh problemov. Sedaj lahko s kvantnimi računalniki uspešno simuliramo preproste molekule in reakcije med njimi. Z nadaljnjim razvojem bo možno simulirati in analizirati molekule, katerih sedaj po standardnih metodah ne moremo pripraviti in proučiti. S tem se bo pospešil razvoj novih zdravil in odkrivanje novih materialov.

Finančna industrija je bila ena prvih industrij, ki je začela izkoriščati velike količine nabranih podatkov, predvsem za določanje cen finančnih instrumentov, kot so opcije. Goldman Sachs uporablja računsko zelo zahtevno metodo, Monte Carlo simulacijo, za napoved tržnih gibanj. Poleg simulacij trga lahko finančne inštitucije uporabljajo to tehnologijo za optimizacijo portfelja, minimizacijo tveganja in optimizacijo posojil.

Kvantni algoritmi so se začeli pojavljati tudi na področju strojnega učenja, strojništva, energije, logistike in še drugod.

3.3 Kvantni algoritmi

Kvantne algoritme delimo na več podskupin, glede na to, na kateri tehnologiji oziroma ideji temeljijo. Nekatero večje podskupine so algoritmi, ki temeljijo na kvantni Fourierovi transformaciji, amplitudnem ojačanju (ang. *amplitude amplification*) in kvantnem sprehodu (ang. *quantum walk*). Vsaka od teh podskupin rešuje različne vrste problemov, od iskanja vrednosti v neurejenem seznamu (Groverjev algoritem) do faktorizacije števila (Shorov algoritem). V tem delu si bomo pogledali delovanje Deutschevega algoritma, ki ga v praksi ne uporabljamo, vendar služi kot dober primer, ki pokaže prednosti kvantnih računalnikov.

Deutschev algoritem je bil eden prvih kvantnih algoritmov, ki je pokazal prednosti in potencial kvantnih računalnikov. Algoritem rešuje sledeči problem. Naj bo f funkcija, ki vzame seznam n bitov in vrne 0 ali 1, torej $f(\{x_0, x_1, \dots, x_{n-1}\}) \rightarrow \{0, 1\}$. Za f predhodno vemo le, da je lahko "konstantna" ali "uravnovešena". Če je f konstantna, vrne 0 ali 1 za vsak vhodni parameter. Funkcija v polovici primerov vrne 0 in v polovici 1, če je uravnovešena. Naš cilj je določiti, ali je funkcija f konstantna ali uravnovešena. Funkcijo f želimo čim manjkrat izvednotiti.

Klasični algoritem lahko ta problem reši v najslabšem primeru v $2^{n-1} + 1$ korakih, kjer en korak pomeni eno izvednotenje funkcije f . Vhodni n -bitni seznam lahko zavzema 2^n različnih vrednosti. V najslabšem primeru moramo torej preveriti točno polovico vseh možnih vrednosti ter še eno dodatno vrednost. Za $n = 2$ bi torej izvednotili $f(0, 0) = 0$ in $f(0, 1) = 0$, tretji izmerek $f(1, 0)$ pa dokončno določi lastnost funkcije: če $f(1, 0) = 0$ je f konstanta, če pa je $f(1, 0) = 1$, je uravnovešena.

Kvantni algoritem lahko dani problem reši le v enem koraku, s čimer smo eksponentno računsko zahtevnost pretvorili na konstantno. Zaradi preprostosti se omejimo na primer $n = 2$. Posplošeni algoritem Deutsch-Jozsa, ki problem reši za poljuben n , je opisan kasneje. Redefinirajmo funkcijo f za uporabo s kvantnim računalnikom, in sicer kot $f(|x\rangle|y\rangle) = |x\rangle|y \oplus f(x)\rangle$. Operacija \oplus je vsota dveh števil, ki jo nato delimo z dva in rezultat je ostanek (na primer $3 \oplus 1 = 0$). Definirajmo neko uravnovešeno funkcijo f kot $f(x) = f(x_0, x_1) = x_0 \oplus x_1$, kjer je x stanje n kubitov, v tem primeru je $n = 2$. Torej velja $f(0, 0) = 0$, $f(0, 1) = 1$, $f(1, 0) = 1$ in $f(1, 1) = 0$. Tu smo kot primer definirali eno izmed možnih uravnovešenih funkcij. Njenega predpisa ne poznamo.

Začetno stanje sistema je $|\psi_0\rangle = |00\rangle_{01} \otimes |1\rangle_2 = [0, 0, 1]^T$. Prva dva kubita smo inicializirali v stanju $|00\rangle$, tretjega pa v stanju $|1\rangle$. Začetno stanje sistema je enako za poljubno funkcijo f . Prvi korak je uporaba Hadamardovih kvantnih vrat, kar da

$$|\psi_1\rangle = H|\psi_0\rangle = H|00\rangle_{01} \otimes H|1\rangle_2 = \frac{1}{2}(|00\rangle + |01\rangle + |10\rangle + |11\rangle)_{01} \otimes \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)_2$$

Tu smo uporabili tudi Hadamardova vrata za dva bita, s čimer smo pretvorili stanje $|00\rangle$ v super-

pozicijo vseh štirih stanj. Uporabimo funkcijo f na $|\psi_1\rangle$, da dobimo

$$\begin{aligned} |\psi_2\rangle &= \frac{1}{2\sqrt{2}}(|00\rangle_{01} \otimes (|0 \oplus 0 \oplus 0\rangle - |1 \oplus 0 \oplus 0\rangle)_2 \\ &\quad + |01\rangle_{01} \otimes (|0 \oplus 0 \oplus 1\rangle - |1 \oplus 0 \oplus 1\rangle)_2 \\ &\quad + |10\rangle_{01} \otimes (|0 \oplus 1 \oplus 0\rangle - |1 \oplus 1 \oplus 0\rangle)_2 \\ &\quad + |11\rangle_{01} \otimes (|0 \oplus 1 \oplus 1\rangle - |1 \oplus 1 \oplus 1\rangle)_2. \end{aligned}$$

Izraz lahko poenostavimo, da dobimo

$$\begin{aligned} |\psi_2\rangle &= \frac{1}{2\sqrt{2}}(|00\rangle_{01} \otimes (|0\rangle - |1\rangle)_2 - |01\rangle_{01} \otimes (|0\rangle - |1\rangle)_2 \\ &\quad - |10\rangle_{01} \otimes (|0\rangle - |1\rangle)_2 + |11\rangle_{01} \otimes (|0\rangle - |1\rangle)_2) = \\ &= \frac{1}{2}(|00\rangle - |01\rangle - |10\rangle + |11\rangle)_{01} \otimes \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)_2 = \\ &= \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)_0 \otimes \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)_1 \otimes \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)_2. \end{aligned}$$

V zadnjem koraku smo stanje prvih dveh kubitov razstavili na produkt posameznih stanj. Če uporabimo Hadamardova vrata na vseh kubitih, dobimo

$$|\psi_3\rangle = H|\psi_2\rangle = |1\rangle_0 \otimes |1\rangle_1 \otimes |1\rangle_2 = |11\rangle_{01} \otimes |1\rangle_2.$$

Če izmerimo stanje prvih dveh kubitov, vidimo, da je verjetnost za stanje $|11\rangle_{01}$ ena, s čimer smo ugotovili, da je funkcija f uravnovešena. V primeru, da bi bila funkcija f konstantna, na primer $f(x_0, x_1) = 1, \forall x_0, x_1$, bi veljalo

$$|\psi_2\rangle = \frac{1}{2\sqrt{2}}(|00\rangle + |01\rangle + |10\rangle + |11\rangle)_{01} \otimes (|1\rangle - |0\rangle)_2.$$

Z uporabo Hadamardovih vrat sedaj dobimo $|\psi_3\rangle = H|\psi_2\rangle = |0\rangle_0 \otimes |0\rangle_1 \otimes |1\rangle_2$. Ko izmerimo rezultat konstantne funkcije, vidimo, da je verjetnost za stanje $|11\rangle_{01}$ enaka nič. Tako lahko z meritvijo kubitov 0 in 1 določimo, ali je funkcija konstantna ali uravnovešena.

Lotimo se sedaj rešitve za splošen problem, torej za poljuben n . Algoritem, ki reši ta problem se imenuje algoritem Deutsch-Jozsa. Opis problema je enak kot za problem $n = 2$, le da tu uporabljamo funkcijo f , ki sprejme n vhodnih kubitov. Funkcijo f v računalniku implementiramo kot linearno transformacijo U_f .

Reševanje problema začnemo z inicializacijo n kubitov v stanju $|0\rangle$ in enega kubita v stanju $|1\rangle$. Naše začetno stanje je torej $|\psi_0\rangle = |0\rangle^{\otimes n} |1\rangle = [0, 0, \dots, 0, 1]^T$, ne glede na f . Sprva uporabimo Hadamardova vrata, kar da

$$|\psi_1\rangle = \frac{1}{\sqrt{2^{n+1}}} \sum_{x=0}^{2^n-1} |x\rangle (|0\rangle - |1\rangle).$$

Tu smo uporabili Hadamardova vrata, ki so posplošena za n bitov, s čimer smo pretvorili stanje $|0\rangle^{\otimes n}$ v superpozicijo vseh 2^n možnih stanj. Za elemente posplošene matrike Hadamardovih vrat velja $(H_n)_{ij} = \frac{1}{\sqrt{2^n}}(-1)^{ij}$. Uporabimo funkcijo f na $|\psi_1\rangle$, da dobimo

$$\begin{aligned} |\psi_2\rangle &= \frac{1}{\sqrt{2^{n+1}}} \sum_{x=0}^{2^n-1} |x\rangle (|0 \oplus f(x)\rangle - |1 \oplus f(x)\rangle) = \\ &= \frac{1}{\sqrt{2^{n+1}}} \sum_{x=0}^{2^n-1} |x\rangle (|f(x)\rangle - |1 \oplus f(x)\rangle) = \\ &= \frac{1}{\sqrt{2^{n+1}}} \sum_{x=0}^{2^n-1} (-1)^{f(x)} |x\rangle (|0\rangle - |1\rangle). \end{aligned}$$

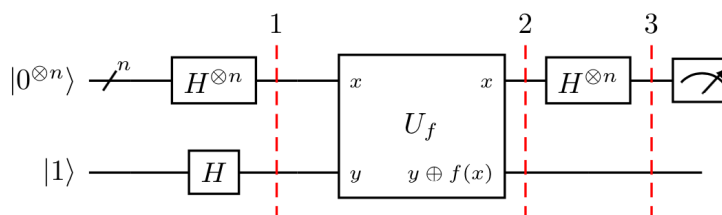
V prvem koraku smo uporabili identiteto $0 \oplus a = a$. Ker je $f(x)$ lahko le $\{0, 1\}$ smo lahko poenostavili zapis v drugem koraku. Na tem mestu lahko opustimo zadnji kubit in nadaljujemo le s prvimi n kubiti, na katerih ponovno uporabimo Hadamardova vrata

$$\begin{aligned} |\psi_3\rangle &= \frac{1}{2^n} \sum_{x=0}^{2^n-1} (-1)^{f(x)} \left(\sum_{y=0}^{2^n-1} (-1)^{x \cdot y} |y\rangle \right) = \\ &= \frac{1}{2^n} \sum_{y=0}^{2^n-1} \left(\sum_{x=0}^{2^n-1} (-1)^{f(x)} (-1)^{x \cdot y} \right) |y\rangle, \end{aligned}$$

kjer je $x \cdot y = x_0y_0 \oplus x_1y_1 \oplus \dots \oplus x_{n-1}y_{n-1}$. Zadnji korak je meritev $|\psi_3\rangle$, pri čemer opazimo, da je verjetnost, da izmerimo stanje $|0\rangle^{\otimes n}$, enaka

$$P(|0\rangle^{\otimes n}) = \left| \frac{1}{2^n} \sum_{x=0}^{2^n-1} (-1)^{f(x)} \right|^2.$$

Verjetnost, da zaznamo to stanje, je 1, če je funkcija f konstantna, in 0, če je uravnovešena. Celoten algoritem lahko zapišemo shematično s kvantnimi vrati, kot je to predstavljeno na sliki 1 [8].



Slika 1. Shema vezja, ki reši Deutsch problem. Na levi strani je prikazano začetno stanje $|\psi_0\rangle = |0\rangle^{\otimes n} |1\rangle$, katero vodimo na Hadamardova vrata. Izhod $|\psi_1\rangle$ uporabimo na funkciji f , da dobimo $|\psi_2\rangle$. U_f so kvantna vrata, ki izvedejo funkcijo f . Rezultat problema dobimo z nadaljnjo uporabo Hadamardovih vrat in meritve stanja [8].

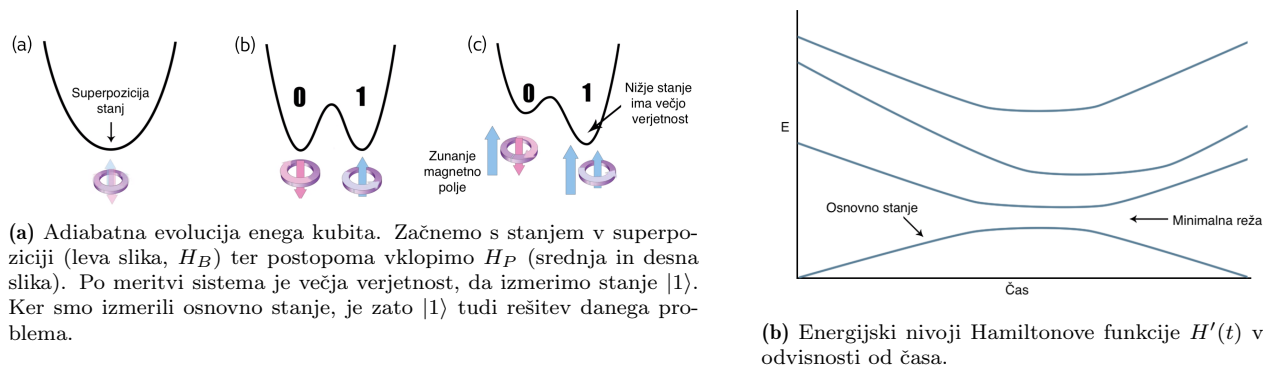
4. Adiatatni kvantni računalnik

Adiatatni kvantni računalnik deluje po načelu adiatatnega teorema. Ta pravi, da če sistem počasi spreminjamo, se bo ta spremembi lahko prilagodil, s čimer se bo spremenila tudi njegova valovna funkcija. Če je sistem sprva v osnovnem lastnem stanju začetne Hamiltonove funkcije, bo po spremembi v osnovnem lastnem stanju končne Hamiltonove funkcije. Bolj preprosto povedano, ta metoda omogoča najti osnovno stanje bolj kompleksnega sistema, pri čemer smo začeli s preprostejšim sistemom. To vrsto računalnika uporabljamo predvsem za reševanje optimizacijskih problemov in problemov vzorčenja. V tem delu se bomo posvetili reševanju optimizacijskih problemov, ki je na klasičnem računalniku zelo potratno. Vsak algoritem, ki ga lahko izvedemo na kvantnem vezju, lahko izvedemo tudi na adiatatnem računalniku, le da je to velikokrat težko izvedljivo oziroma veliko manj učinkovito.

Vsak problem, ki ga želimo rešiti z adiatatnim računalnikom, moramo najprej zapisati s Hamiltonovo funkcijo (H_P), katerega osnovno stanje ustreza rešitvi problema. Nato pripravimo osnovno stanje sistema s preprosto Hamiltonovo funkcijo (H_B), ki deluje na n kubitov. Zadnji korak je, da počasi vključimo H_P in izključujemo H_B :

$$H'(t) = \left(1 - \frac{t}{T}\right) H_B + \frac{t}{T} H_P,$$

kjer je H' skupna Hamiltonova funkcija in T čas vključevanja. Začetno in končno lastno stanje se razlikujeta, vendar sta obe stanji osnovni stanji pripadajočih Hamiltonovih funkcij. Po adiabatni evoluciji, katero ponazarja slika 2a, izmerimo stanje sistema oziroma kubitov, kar predstavlja rešitev našega problema. Adiabatna evolucija je proces, ko pustimo sistemu, da se časovno razvija po Schrödingerjevi enačbi, s časovno odvisno Hamiltonovo funkcijo.



Slika 2. Lastnosti adiabatnega kvantnega računalnika. Sliki prilagojeni po [9].

Čas vključitve je zelo odvisen od lastnosti Hamiltonove funkcije ter posledično od našega problema. Čas evolucije je omejen z naslednjo enačbo:

$$T = \mathcal{O}\left(\frac{1}{\min g(t)^2}\right)$$

kjer je $g(t)$ razlika v energiji osnovnega in prvega vzbujenega stanja sistema [4]. Odvisnost $g(t)$ ponazarja slika 2b. V primeru, da celoten proces izvedemo prehitro, je velika verjetnost, da na koncu ne bomo več v osnovnem stanju H_P , saj smo sistemu dovedli dovolj energije, da preide v višje stanje. Na razliko v energiji in posledično na stabilnost sistema ter pravilnost rešitve vpliva tudi število kubitov in zunanji vplivi. Slednji so lahko previsoka temperatura sistema in motnje, kot na primer elektromagnetno sevanje. Večje, kot je število kubitov, manjša je energijska reža [4][5].

4.1 Optimizacija portfelja

Eno od vprašanj, ki se pojavi, je, kako zapisati problem s Hamiltonovo funkcijo, tako da bo njegovo osnovno stanje rešitev problema. Za boljše razumevanje si to pogledjmo na primeru iskanja optimalnega portfelja.

Adiabatne kvantne računalnike lahko uporabimo za iskanje globalnih minimumov poljubne funkcije več spremenljivk, kar je težko izvedljivo na klasičnem računalniku zaradi lokalnih minimumov. To je posledica tega, da smo na začetku začeli v globalnem minimumu oziroma osnovnem stanju. Optimizacijske oziroma minimizacijske probleme zelo pogosto srečamo v fiziki, inženirstvu in računalništvu, v tem delu pa bomo optimizirali finančni portfelj.

Naloga investitorja je izbrati delnice ali druge finančne instrumente ter se odločiti, kdaj jih kupiti in prodati ter koliko jih kupiti. Zaradi velikega števila delnic in možnih transakcij je kvantna optimizacija primerna metoda za reševanje problema. Naš cilj je zasnovati algoritem, s katerim bomo določili optimalen načrt nakupa in prodaje delnic, ki upošteva stroške in minimizira tveganje.

Naj bo N število delnic in \bar{w}_t N -dimenzionalni vektor, kjer je $\bar{w}_{t,n} \in [0, 1]$. $\bar{w}_{t,n}$ predstavlja, kolikšno vrednost n -te delnice imamo ob času t . Če imamo dve delnici s ceno 100 je $\bar{w}_{t,n} = 200$. Za čas velja $t = t_z, t_z + 1, \dots, t_k - 1, t_k$, kjer je t_z začetni čas in t_k končni čas. Število časovnih členov je N_t . Naj bo μ_t predviden dobiček delnice ob času t in Σ_t kovarianca med delnicami. μ_t je N -dimenzionalni vektor, Σ_t pa $N \times N$ matrika. Finančno tveganje je povezana z volatilitvijo dobička,

kar lahko merimo s standardno deviacijo ali kovarianco (Σ_t). Kovarianca med dvema delnicama nam pove, kako se bo odzvala vrednost delnice na spremembo v vrednosti druge delnice. V splošnem želimo, da sta delnici nasprotno korelirani, saj se tako zavarujemo pred izgubami. Vrednosti μ_t in Σ_t lahko aproksimiramo iz zgodovinskih podatkov. Končni rezultat algoritma je set $\{\bar{\omega}_{t_z}, \dots, \bar{\omega}_{t_k}\}$, ki predstavlja optimalno strategijo. Celoten dobiček znaša

$$\sum_{t=t_z}^{t_k} \mu_t^T \bar{\omega}_t,$$

celotno tveganje pa

$$\frac{1}{2} \sum_{t=t_z}^{t_k} \bar{\omega}_t^T \Sigma_t \bar{\omega}_t.$$

Faktor $\frac{1}{2}$ navajamo zaradi konvencije. Pogosto želimo tudi omejiti maksimalno količino sredstev (K), ki jih želimo porabiti

$$\sum_{n=1}^N \bar{\omega}_{n,t} = K, \forall t.$$

Poenostavimo zapis, tako da normaliziramo $\bar{\omega}_t$ glede na celotno investicijo

$$\omega_{n,t} = \frac{\bar{\omega}_{n,t}}{K},$$

tako da je vsota vseh deležev delnic enaka 1 za vsak t . $\omega_{n,t}$ predstavlja delež delnice n glede na celoten portfelj ob času t . Definirajmo minimizacijsko funkcijo H , katere minimum predstavlja optimalno rešitev

$$H = \sum_{t=t_z}^{t_k} \left(-\mu_t^T \omega_t + \frac{\gamma}{2} \omega_t^T \Sigma_t \omega_t + \rho (\mathbf{1}^T \omega_t - 1)^2 \right).$$

Manjši H je posledica večjega dobička (prvi člen), manjšega tveganja (drugi člen) in porabe celotnega proračuna (tretji člen). γ in ρ sta faktorja, ki ju nastavi uporabnik in predstavljata stopnjo tveganja in stopnjo kaznovanja pri porabi sredstev. Opazimo, da je Hamiltonova funkcija diagonalna v času. To pomeni, da je celotna rešitev problema vsota rešitev ob posameznih časovnih korakih. V splošnem imamo lahko tudi časovno korelirane člene, pri čemer navedena lastnost ne velja.

Problem otežimo, če dodamo stroške transakcije. Ti stroški ponavadi predstavljajo nek delež prodajne ali nakupne cene delnice, kar lahko zapišemo kot

$$\sum_{t=t_z}^{t_k} \nu_t |\Delta \omega_t| = \sum_{t=t_z}^{t_k} \sum_{n=1}^N \nu_{n,t} |\omega_{n,t+1} - \omega_{n,t}|, \quad (2)$$

kjer je $\nu_{n,t}$ delež transakcije, ki ustreza prodaji delnice n ob času t . Z uvedbo tega člena dobimo povezavo med časoma t in $t+1$.

Absolutna vrednost nas omejuje pri določenih kvantnih optimizacijskih algoritmih, zato jo lahko s Taylorjevim razvojem razpišemo do parabole, s čimer dobimo polinomsko odvisnost Hamiltonove funkcije. $\nu_{n,t}$ lahko v večini primerov zamenjamo s konstanto λ . Hamiltonovo funkcijo zdaj zapišemo kot

$$H = \sum_{t=t_z}^{t_k} \left(-\mu_t^T \omega_t + \frac{\gamma}{2} \omega_t^T \Sigma_t \omega_t + \lambda (\Delta \omega_t)^2 + \rho (\mathbb{1}^T \omega_t - 1)^2 \right). \quad (3)$$

Celoten profit strategije znaša

$$\sum_{t=t_z}^{t_k} \mu_t^T \omega_t - \lambda (\Delta \omega_t)^2.$$

Sedaj smo zapisali minimizacijsko funkcijo v obliki Hamiltonove funkcije. Implementacija le-tega v kvantnem računalniku je podana v naslednjem poglavju [2][3].

4.2 Kvantni računalnik D-Wave

Kvantni računalnik podjetja D-Wave (*quantum annealer*) deluje po načelu kvantnega adiabatskega računalnika. Procesna enota vsebuje mrežo superprevodnih zank, ki tvorijo kubite. Nanje lahko vplivamo s spreminjanjem magnetnega polja. Na začetku vseh n kubitov postavimo v superpozicijo stanja $|0\rangle$ in $|1\rangle$. Tako dobimo sistem, kjer so vsa stanja (2^n stanj) enako verjetna. S spreminjanjem zunanje magnetnega polja lahko spreminjamo verjetnost, da posamezen bit zavzame določeno stanje [5].

Glavno moč računalnika dobimo, ko lahko nadziramo interakcije med posameznimi kubitami. Le-to naredimo s spojki (ang. *coupler*), ki povežejo dva kubita, tako da po meritvi oba končata v istem ali v nasprotnem stanju. Uporabnik lahko spreminja moč sklopitve med dvema kubitoma.

Kvantno mehanska spojka preplete kubita. Takrat oba kubita obravnavamo kot en objekt s štirimi stanji. Verjetnost za posamezno stanje določimo z magnetnim poljem in močjo sklopitve.

Med procesom adiabatske evolucije počasi vklopimo polje in spojke, s čimer definiramo Hamiltonovo funkcijo sistema. Na koncu izmerimo stanja kubitov ter tako določimo osnovno stanje. Celoten proces traja okoli nekaj mikrosekund.

Naslednji del reševanja finančne optimizacije je zapisati H z interakcijami med kubitami. Kvantni računalnik D-Wave za zapis interakcij med kubitami uporablja Isingov model. Hamiltonovo funkcijo računalnika zapišemo kot

$$H_{Ising} = \left(1 - \frac{t}{T}\right) \sum_{i=0}^n \sigma_x^{(i)} + \frac{t}{T} \left(\sum_{i=0}^n h_i \sigma_z^{(i)} + \sum_{i=0}^n \sum_{j=0}^i J_{ij} \sigma_z^{(i)} \sigma_z^{(j)} \right),$$

kjer sta $\sigma_z^{(i)}$ in $\sigma_x^{(i)}$ Paulijevi matriki, ki delujeta na kubit q_i . h_i in J_{ij} sta zunanje magnetno polje in moč sklopitve med kubitoma. Prvi člen je začetna Hamiltonova funkcija, katere osnovno stanje so vsi kubiti v superpoziciji $|0\rangle$ in $|1\rangle$. Drugi člen predstavlja končno Hamiltonovo funkcijo. Običajno prehod med začetno in končno Hamiltonovo funkcijo ni linearen, kot je prikazano zgoraj, ampak ga določata dve funkciji, ki ju določimo za optimalno delovanje sistema.

Lastna stanja Hamiltonove funkcije iz enačbe (3) so vse kombinacije portfelja. Posamezno stanje predstavlja vektor ω . V enačbi (3) se pojavljajo linearni (npr. ω_t) in kvadratni členi (npr. $\omega_t \omega_{t+1}$). Faktorje pred linearnimi členi enačimo s h_i , tiste pred kvadratnimi pa z J_{ij} . Zato, da smo lahko zapisali problem v obliki Isingovega modela, smo pretvorili absolutno vrednost iz enačbe (2) v polinom. Računalniški model lahko vključuje tudi višje polinomske člene, ki predstavljajo sklopitve med več kubitami. S tem smo prepisali naš problem v obliko, primerno za kvantni adiabatski računalnik [5].

4.3 Variacijski iskalec lastnih stanj

Optimizacijski problem lahko rešimo tudi na hibriden način, ki združuje koncepte iz adiabatskega modela in modela vrat. Variacijski iskalec lastnih stanj (ang. *variational quantum eigensolver*) uporablja variacijsko metodo, ki sistematično pretvori neko začetno stanje v osnovno stanje sistema. To naredi z uporabo parametriziranega vezja ali tako imenovane variacijske forme, ki jo zapišemo kot linearno transformacijo $U(\theta)$. Z variacijsko formo pretvorimo začetno stanje $|\psi\rangle$ v stanje $U(\theta)|\psi\rangle = |\psi(\theta)\rangle$. Skozi iterativen proces želimo najti optimalno vrednost vektorja θ , da zagotovimo naslednjo enakost

$$\langle \psi(\theta) | H | \psi(\theta) \rangle \approx E_0.$$

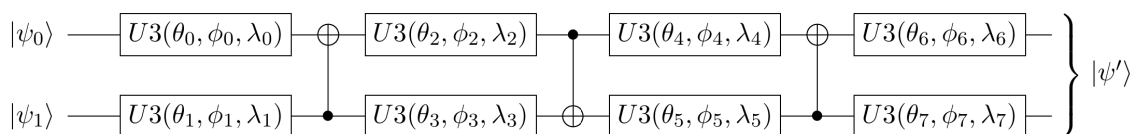
V idealnem primeru želimo, da je $|\psi(\theta)\rangle = |\psi_0\rangle$, kjer je $|\psi_0\rangle$ osnovno stanje H .

Pri izbiri variacijske forme moramo biti pozorni na dve stvari. Želimo, da n kubitov variacijske forme opiše vsa možna stanja sistema, hkrati pa želimo, da ta vsebuje čim manj parametrov. Zaradi preprostosti se sedaj osredotočimo le na prvi cilj.

Zapišimo variacijsko formo za en kubit. Uvedemo jo kot kvantna vrata U3, ki jih parametriziramo s θ , ϕ in λ ter jih zapišemo kot

$$U3(\theta, \phi, \lambda) = \begin{bmatrix} \cos(\frac{\theta}{2}) & -e^{i\lambda} \sin(\frac{\theta}{2}) \\ e^{i\phi} \sin(\frac{\theta}{2}) & e^{i\lambda+i\phi} \cos(\frac{\theta}{2}) \end{bmatrix}. \quad (4)$$

Vrata U3 lahko izvedejo poljubno transformacijo na enem kubitju do globalne faze natančno. Univerzalno variacijsko formo za dva kubita zapišemo kot vezje osmih vrat U3 in treh vrat CNOT. Formo smo parametrizirali s 24 parametri, podana pa je z vezjem na sliki 3. Variacijsko formo lahko sestavimo za poljubno število kubitov.



Slika 3. Shema vezja univerzalne variacijske forme za dva kubita. U3 je transformacija, ki jo podaja enačba (4). Označe, ki povezujejo vodoravni liniji, predstavljajo CNOT vrata [8].

Sedaj, ko smo določili variacijsko formo, moramo še poiskati stanje z najmanjšo energijo. Najpreprostejša metoda je gradientni spust oziroma Newtonova metoda, kjer spremenimo vsak parameter, tako da dobimo največjo razliko v energiji. Algoritem lahko hitro najde lokalne ekstreme, vendar se v njih pogosto ustavi ter tako ne najde globalnega minimuma. Zato je uporaba te metode odsvetovana.

Bolj primeren je optimizator SPSA (ang. *Simultaneous Perturbation Stochastic Approximation optimizer*). SPSA aproksimira gradient le z dvema klicema funkcije oziroma meritvama sistema. To naredi tako, da sočasno naključno perturbira vse parametre, kar je v nasprotju z gradientnim spustom, kjer spreminjamo vsak parameter individualno. Poleg SPSA obstajajo tudi drugi optimizatorji, kot sta SLSQP in COBYLA [8].

5. Implementacija kubitov in kvantnega vezja

V prejšnjih poglavjih smo predpostavili, da imamo enega ali več kubitov ter da lahko na njih delujemo s kvantnimi vrati, ne da bi omenili, kako to izvesti v praksi. Kubit lahko reprezentiramo z elektronom, fotonom, ionom, superprevodnimi zankami ali drugimi sistemi, z njim pa lahko interagiramo z različnimi metodami. V tem delu si bomo pogledali dva najbolj uporabljena načina, kvantni računalnik na superprevodne zanke (Google in IBM) in ujete ione (IonQ).

5.1 Superprevodni kvantni računalnik

Osnovni nosilci naboja v superprevodniku so pari elektronov, tako imenovani Cooperjevi pari, in ne posamezni elektroni, kot je to v navadnem prevodniku. Cooperjevi pari so bozoni, saj imajo celoštevilski skupni spin. Za razliko od fermionov lahko bozone tako ohladimo, da zasedajo le en kvantni nivo, čemur pravimo Bose-Einsteinova kondenzacija. Tako lahko vse pare obravnavamo kot en objekt ter ga zapišemo le z eno valovno funkcijo. To nam tudi omogoča merjenje kvantnih efektov na makroskopskem nivoju. Na primer, skozi superprevodno zanko prehaja le diskretno število kvantov magnetnega pretoka.

Superprevodni kvantni računalnik uporablja superprevodne zanke za zapis kubita. Zanko obravnavamo kot preprosto vezje s kondenzatorjem in tuljavo, ki ima pri nizkih temperaturah (okoli 0,1 K) diskretne energijske nivoje. Ker želimo, da so energijske razlike med stanji različne, v zanko dodamo nelinearen induktor, tako imenovan Josephsonov spoj. Te tri elemente lahko vežemo na različne načine, da tvorimo različne tipe kubitov. V nabojnem kubitju različni energijski nivoji pripadajo različnim številom Cooperjevih parov v superprevodniku. Energijski nivo v pretočnem kubitju je sorazmeren številu kvantov magnetnega pretoka, ujetega v superprevodni zanki. Na pretočni kubit lahko gledamo tudi bolj klasično in si stanja predstavimo kot tok, ki teče v desno ali v levo smer po superprevodni zanki. Kubita $|0\rangle$ in $|1\rangle$ lahko definiramo kot

$$|0\rangle = \frac{1}{\sqrt{2}}(|\circ\rangle + |\ominus\rangle) \quad |1\rangle = \frac{1}{\sqrt{2}}(|\circ\rangle - |\ominus\rangle).$$

Kubit je lahko v poljubni superpoziciji zgornjih stanj, kar dosežemo s pulzi mikrovalov pri določeni frekvenci, s tem pa smo dobili način za uvedbo kvantnih vrat.

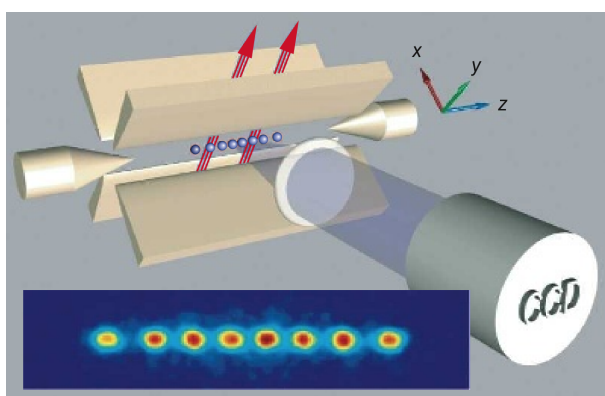
Končno stanje kubita izmerimo z magnetometrom SQUID, ki omogoča zelo natančno merjenje magnetnega pretoka.

Z mikrovalovi nadzorujemo stanja kubitov in tvorimo kvantna vrata [6].

5.2 Kvantni računalnik na ujete ione

Kubit v kvantnem računalniku na ujete ione predstavlja ion. Elektronsko stanje iona določa stanje kubita.

Prvi korak pri vzpostavitvi sistema je, da uparimo atome (na primer kalcijeve ali iterbijeve), ki jih obstreljujemo z elektroni, tako da atome ioniziramo. Z električnim poljem ione vodimo do ionskih pasti. S statičnim električnim poljem ne moremo ustvariti stabilnega potenciala za nabit delec, zato uporabimo električno in magnetno polje (Penningova past) ali pa časovno spreminjajoče električno polje (Paulova past). Na koncu ione z laserjem ohladimo v osnovno stanje, kar predstavlja kubitno stanje $|0\rangle$. Ujeti ioni morajo biti v vakuumu, da ne interagirajo z okoliško snovjo. Za obratovanje ne potrebujemo tako nizkih temperatur kot pri superprevodnih zankah. Celotno aparaturu prikazuje slika 4.



Slika 4. Shema kvantnega računalnika ujetih ionov [10]. Modre kroglice predstavljajo ujete ione oziroma kubite, ki so obdani z elektrodami (rjava barva). Te stabilizirajo ione v vakuumu s spreminjajočim se električnim poljem. Desno spodaj je CCD kamera, ki s pomočjo laserja izmeri stanja kubitov. Levo spodaj je slika ionov, posneta s CCD kamero.

Obstajata dve glavni vrsti kubitov ujetih ionov: optični in hiperfini kubiti. Le-ti se razlikujejo v energijski razliki med stanjema $|0\rangle$ in $|1\rangle$. Za prehod optičnega kubita uporabljamo laserje z vidno svetlobo ($\nu \approx 10^{14} - 10^{15}$ Hz). Pri hiperfinih kubitih uporabljamo mikrovalove s frekvenco

$\nu \approx 10^9 - 10^{10}$ Hz. Z uporabo laserjev ali mikrovalov lahko zelo natančno nadziramo stanje kubitov in interakcije med njimi, s čimer lahko implementiramo kvantna vrata.

Stanje sistema izmerimo, tako da na ione svetimo z laserjem točno določene frekvence, pri kateri eno od stanj (npr. $|1\rangle$) odda svetlobo, drugo pa ne. S CCD kamero nato merimo prepuščeno svetlobo ter določimo končno stanje sistema [6][7].

6. Zaključek

Kvantno računalništvo je še zelo odprto in neustaljeno področje. Trenutno se lotevamo realizacije kvantnega računalnika na več kot 20 različnih načinov. Skupni cilj vseh je enak: izdelati računalnik, ki lahko izvaja osnovne operacije na kubitih in ima poljubno število dolgo obstojnih kubitov. Z nadaljnjim raziskovanjem bomo videli, katera metoda bo prevladala. V vmesnem času se že razvija in do neke mere tudi uporablja algoritme na področju kvantne simulacije, strojnega učenja, faktORIZACIJE in optimizacije. Sprva bomo uporabo kvantnih računalnikov videli v farmacevtski, kemijski, avtomobilski in finančni industriji. Na primer v farmacevtski industriji bi lahko s kvantno simulacijo zelo zmanjšali čas razvoja novih zdravil. Primer uporabe v finančni industriji je poleg optimizacije portfelja tudi zmanjševanje tveganj investicij in optimizacija posojil. Kvantni računalniki postajajo tudi bolj dostopni javnosti s platformam, kot sta Qiskit (IBM) in Quantum Playground (Google).

LITERATURA

- [1] A. Ramšak, *Kvantna mehanika*, Založba Univerze (2021).
- [2] S. Mugel, C. Kuchkovsky, E. Sanchez, S. Fernandez-Lorenzo, J. Luis-Hita, E. Lizaso in R. Orus, *Dynamic Portfolio Optimization with Real Datasets Using Quantum Processors and Quantum-Inspired Tensor Networks*, (2020).
- [3] S. Palmer, S. Sahin, R. Hernandez, S. Mugel in R. Orus, *Quantum Portfolio Optimization with Investment Bands and Target Volatility*, (2021).
- [4] T. Albash in D. Lidar, *Adiabatic Quantum Computing*, (2016).
- [5] M. Ohzeki in H. Nishimori, *Quantum Annealing: An Introduction and New Developments. Journal of Computational and Theoretical Nanoscience*, (2010).
- [6] National Academies of Sciences, Engineering, and Medicine, *Quantum Computing: Progress and Prospects*, The National Academies Press, Washington, DC, (2019).
- [7] J. Hui, *How to build a Quantum Computer with Trapped Ions*, Medium (2019).
- [8] *Learn Quantum Computation Using Qiskit*, IBM, (2020).
- [9] *D-Wave System Documentation*, D-Wave.
- [10] R. Blatt in D. Wineland, *Entangled States of Trapped Atomic Ions*, Nature. 453. 1008 15. 10.1038/nature07125 (2008).