Tampereen yliopisto

Eetu Honkanen

KVANTTILASKENTA

Kandidaatin tutkielma
Tekniikan ja luonnontieteiden tiedekunta
Tarkastaja: Prof. Tapio Rantala
Toukokuu 2021

TIIVISTELMÄ

Eetu Honkanen: Kvanttilaskenta
Kandidaatin tutkielma
Tampereen yliopisto
Tekniikan ja luonnontieteiden kandidaattiohjelma
Toukokuu 2021

Kvanttimekaniikkaan pohjautuvan kvanttilaskennan avulla näyttää olevan mahdollista ratkaista joitain ongelmia merkittävästi klassista laskentaa nopeammin. Tämän kandidaatintyön tavoite on selvittää, kuinka kvanttilaskennalla saavutetaan etu laskennallisten ongelmien ratkaisemisessa, ja millaisissa sovelluksissa sitä voidaan hyödyntää.

Työ koostuu kirjallisuustutkimuksesta sekä simulaatiolla ja kvanttitietokoneella toteutetusta laskennallisesta osasta. Kirjallisuustutkimusosassa esitellään kvanttilaskennan kannalta oleellisilta osin kvanttimekaniikan teoriaa. Työstä selviää, kuinka kvanttisuperpositio, superpositiotilojen interferenssi ja lomittuminen aiheuttavat kvantti-informaation erilaisen luonteen klassiseen informaatioon nähden. Työssä tarkastellaan laskentaa kvanttipiirimallilla ja käydään läpi kvanttilaskentaprosessin rakentuminen yksittäisten kvanttibittien porttioperaatioista monimutkaisemmiksi kvanttialgoritmeiksi. Löydetyillä kvanttialgoritmeilla voidaan etsiä tietoa, ratkaista lineaarisia yhtälöryhmiä ja simuloida kvanttimekaanisia systeemejä merkittävästi moderneja supertietokoneita tehokkaammin. Parhaimmillaan nopeudenlisäys on eksponentiaalinen.

Tekniikan kehitys on mahdollistanut viime vuosikymmeninä ensimmäisten kvanttitietokoneiden rakentamisen. Toteutetut kubitit ovat hyvin alttiita ympäristön vuorovaikutuksen aiheuttamille virheille, mikä on suurin este tehokkaiden kvanttitietokoneiden kehitykselle. Täysin virheensietoisella kvanttitietokoneella olisi mahdollista murtaa nykyisiä salausjärjestelmiä, simuloida monimutkaisia kemiallisia reaktioita ja ratkaista hyvin kompleksisia optimointiongelmia, jotka ovat klassisin keinoin saavuttamattomissa. Virheensietoinen kvanttitietokone olisi mullistava keksintö, jolle olisi sovelluksia hyvin monella tieteenalalla kryptografiasta biotekniikkaan.

Työn lyhyessä laskennallisessa osassa toteutettiin yhden iteraation Groverin hakualgoritmi kolmella kubitilla IBM Q Experience -alustalla. Algoritmi ajettiin 1000 kertaa sekä simulaattorilla että kvanttitietokoneella. Vaikka kvanttitietokoneella esiintyi laskennassa huomattavaa virhettä, suoriutui se tehtävästä paremmin kuin vastaavalla klassisella algoritmilla olisi mahdollista. Saaduilla tuloksilla voidaan osoittaa, että nykyisin avoimesti saatavilla olevalla kvanttitietokoneella voidaan suorittaa onnistuneesti yksinkertaisia kvanttimekaanisiin ilmiöihin pohjautuvia laskutoimituksia.

Avainsanat: kvanttilaskenta, kvanttitietokone, kvantti-informaatio, kvanttialgoritmi, kvanttipiiri

ALKUSANAT

Tahdon kiitttää ohjaajaani Tapio Rantalaa mahdollisuudesta mielenkiintoiseen aihevalintaan sekä kaikesta saamastani avusta työn aikana. Haluan myös kiittää perheeltä ja ystäviltäni saamastani kannustavista kommenteista työn aikana.

Tampereella, 16. toukokuuta 2021

Eetu Honkanen

SISÄLLYSLUETTELO

1. JOHDANTO

Viime vuosisadan alussa kvanttifysiikan löytäminen johti transistorin keksintään, jota voidaan pitää nykyisten tietotekniikan perustana. Nykyisten tietokoneiden voidaankin sanoa olevan kvanttifysiikan sovellus. Puolijohdeteknologian kehitys on ollut viimeisten vuosikymmenten aikana nopeaa, mikä on mahdollistanut suorituskyvyn parantamisen komponenttien kokoa pienentämällä. Tässä kehityksessä on kuitenkin tullut raja vastaan, sillä atomaarisessa mittakaavassa hallitsevan kvanttimekaniikan epädeterministinen luonne aiheuttaa virhettä logiikkapiireissä.

Kvanttimekaniikan epädeterministisen luonteen voi nähdä myös mahdollisuutena. 1980-luvulla esitettiin ensimmäistä kertaa ajatus, että kvanttimekaniikan ilmiöitä voisi hyödyntää laskennassa. Arveltiin, että kvanttitietokone voisi tehdä jotain mikä olisi tavanomaiselle tietokoneelle mahdotonta. Myöhemmin ajatuksen ympärille on rakentunut oma tieteenalansa, kvantti-informaatiotiede. Viime vuosikymmeninä kvanttilaskentaa on myös päästy testaamaan fyysisesti ensimmäisillä kvanttitietokoneilla.

Kvanttilaskenta on edelleen laajentuva tutkimuksen kohde. Työn tavoitteena on tutkia kirjallisuudesta, kuinka kvanttilaskennalla voidaan saavuttaa etu klassiseen laskentaan nähden ja mitä sillä voidaan saavuttaa sekä osoittaa kvanttilaskennan tehokkuus käytännön demonstraatiolla.

Luvussa 2 tarkastellaan kvanttilaskennan kannalta oleellista kvanttimekaniikan formalismia ja ilmiöitä. Luvussa 3 perehdytään kvanttimekaanisen systeemin hyödyntämiseen informaation tallentamisessa. Luvuissa 4 ja 5 käydään läpi kvanttimekaanisen laskennan prosessia. Luvussa 6 käsitellään kvanttitietokoneen fyysistä toteutusta sekä kvanttitietokoneen mahdollisia tulevaisuuden sovelluksia. Luvussa 7 suoritetaan yksinkertainen laskutoimitus kvanttitietietokoneella ja kvanttilaskennan klassisena simulaationa. Lopuksi luku 8 kokoaa yhteen tärkeimmät havainnot ja johtopäätökset.

2. KVANTTILASKENNAN KVANTTIMEKANINEN TAUSTA

Klassisen tietokoneen toiminta on esitettävissä klassisella fysiikalla. Kvanttitietokoneen toiminta perustuu sen sijaan kvanttimekaniikkaan, minkä takia sen käsittely vaatii kvanttimekaniikan perusymmärrystä.

2.1 Tila-avaruus

Kvanttimekaanisen systeemin aaltofunktion kaikki mahdolliset tilat voidaan esittää vektorina kompleksisessa Hilbertin avaruudessa, niin kutsutussa tila-avaruudessa. Tila-avaruuden dimensiot riippuvat tarkasteltavasta systeemistä. Tilavektori käsittää kaiken mahdollisen informaation systeemin tilasta. [1]

Kvanttimekaniikassa käytetään usein bra-ket-notaatiota. Tilavekorin kertomisella skalaarilla ei ole fysikaalista merkitystä, jolloin tilat $|v\rangle$ ja $|w\rangle$ vastaavat toisiaan [2].

$$|v\rangle \sim |w\rangle \quad |v\rangle = \lambda\,|w\rangle \quad \lambda \in \mathbb{C}. \tag{2.1}$$

Tiloja käsitellään yleisesti yksikkövektoreina.

Jokaista mitattavaa suuretta, observaabelia α vastaa Hilbertin avaruudessa lineaarinen hermiittinen operaattori A. Operaattorin A omianisarvoyhtälö

$$Au_n = a_n u_n \tag{2.2}$$

määrittelee suureen ominaisarvot a_n ja niitä vastaavat ominaistilat u_n. Kun observaabelia mitataan, voidaan mittaustuloksina saada ainoastaan ominaisarvoja, jolloin mitattaessa systeemi siirtyy ominaisarvoa vastaavalle ominaistilalle. Observaabelit ovat tila-avaruudessa hermiittisiä operaattoreita, minkä takia niiden ominaisarvot reaalisia ja sopivat mittaustuloksiksi. [3, s. 21] Hermiittiselle matriisille A ja sen konjugaattitranspoosille $A^\dagger$ pätee $A = A^\dagger$.

Systeemin todennäköisyys siirtyä mitattaessa tietylle ominaistilalle määritellään tila-avaruudessa sisätulon avulla. Kahden tilavektorin ψ_1 ja ψ_2 sisätulo on kompleksiluku, jota kutsutaan todennäköisyysamplitudiksi ϕ. Varsinainen fysikaalinen merkitys on todennäköisyysampli-

tudin itseisarvon neliöllä, joka vastaa siirtymän todennäköisyyttä P. [3, s. 17]

$$\phi = \langle \psi_1 | \psi_2 \rangle \quad \phi \in \mathbb{C} \tag{2.3}$$

$$P = |\phi|^2 \qquad P \in \mathbb{R} \tag{2.4}$$

Systeemin ominaistilat muodostavat Hilbertin avaruuden ortogonaalisen kannan, minkä takia kaikki mahdolliset superpositiot voidaan esittää ominaistilojen lineaarikombinaatioina [4]

$$|\Psi\rangle = \sum_{n=1}^{n} a_n |\alpha_n\rangle \,. \tag{2.5}$$

Kaavassa 2.5 kertoimet a_n ovat ominaistilojen α_n todennäköisyysamplitudeja. Kerrointen neliöt siis vastaavat suoraan kunkin ominaistilan esiintymisen todennäköisyyttä, kun systeemi mitataan. Systeemin on mitattaessa siirryttävä jollekin ominaistilalleen, eli

$$\sum_{n=1}^{n} |a_n|^2 = 1. \tag{2.6}$$

Tila-avaruus riippuu tarkasteltavasta systeemistä. Kahden systeemin tapauksesssa on selvää, että mahdollisia tiloja on huomattavasti suurempi määrä kuin yksittäisen systeemin tapauksessa. Kahden systeemin $|\psi_A\rangle$ ja $|\psi_B\rangle$ muodostama tila määritellään tilavektoreiden tensoritulona [3, p. 50]

$$|\psi_A \otimes \psi_B\rangle = |\psi_A\rangle \otimes |\psi_B\rangle = |\psi_A \psi_B\rangle \,. \tag{2.7}$$

Myös tila-avaruuden kanta voidaan määrittää tensoritulon avulla osasysteemien tila-avaruuksien kannoista. Systeemit, joiden tila-avaruudet ovat $\mathcal{H}_A$ ja $\mathcal{H}_B$, muodostavat yhdessä systeemin, jonka tila-avaruus $\mathcal{H}_{AB}$ saadaan tensoritulona [3, s. 50]

$$\mathcal{H}_{AB} = \mathcal{H}_A \otimes \mathcal{H}_B. \tag{2.8}$$

Muotoa 2.7 olevat tilat muodostavat vain pienen tilan osan tila-avaruuden $\mathcal{H}_{AB}$ mahdollisista tiloista, sillä suuri osa tiloista on lomittuneita. [3, s. 50]

2.2 Lomittuminen

Usean kvanttimekaanisen systeemin tilannetta, jossa yhden systeemin tila vaikuttaa erottamattomasti muiden systeemien tiloihin, kutsutaan lomittumiseksi. Einstein *et al.* 1935 esitti ajatuskokeen [5], jossa on kaksi kvanttimekaanista systeemiä. Aluksi ne ovat interaktiossa keskenään, minkä jälkeen systeemit erotetaan toisistaan. Interaktiossa ollessaan systeemit ovat vaikuttaneet toistensa tiloihin. Myöhemmin toisen osasysteemin tullessa mitatuksi tullaan määrittäneeksi myös toisen osasysteemin tila. Ilmiö rikkoo lokaalisen reaalisuuden periaatetta, minkä takia sitä pidettiin osoituksena kvanttimekaniikan teorian epätäydellisyydestä.

Myöhemmin lokaalisen reaalisuuden on osoitettu olevan yhteensopimaton konsepti kvanttimekaniikan teorian kanssa. Systeemin tila on aidosti ennalta-arvaamaton ja ajatuskokeessa ensin mitattu systeemi vaikuttaa välittömästi toisen systeemin tilaan. [6] Ilmiö on todennettu useasti kokeellisesti, eikä välimatkalla näyttäisi olevan vaikutusta. Viime vuosina lomittuneita fotoneita on mitattu suurilla, yli 1200 km:n etäisyyksillä. [7]

2.3 Kvanttimekaanisen systeemin aikaevoluutio

Schrödingerin yhtälö 2.9 on osa kvanttimekaanisen teorian perustaa ja se määrittelee kaikkia kvanttimekaanisia systeemeitä. [8] Schrödingerin yhtälö voidaan kirjoittaa muodossa

$$i\hbar\frac{\partial}{\partial t}\left|\psi(t)\right\rangle = \hat{H}\left|\psi(t)\right\rangle, \tag{2.9}$$

jossa $|\psi(t)\rangle$ on systeemin tila hetkellä t, $\hbar$ on redusoitu Planckin vakio ja $\hat{H}$ on systeemin kokonaisenergiaa vastaava Hamiltonin operaattori. Kvanttimekaanisen systeemin tila ei ole koskaan staattinen, vaan se on jatkuvassa muutoksessa [1, s. 12]. Muutosta kutsutaan kvanttimekaanisen systeemin aikaevoluutioksi.

Systeemin sanotaan olevan stationäärinen, kun sen potentiaali on ominaistilallaan ajasta riippumaton. Tällöin myös systeemin Hamiltonin operaattori on ajan suhteen vakio, $\hat{H} = E$. Kun stationaarisen systeemin tila tunnetaan ajanhetkellä $t = 0$, saadaan Schrödingerin yhtälöstä tila hetkellä t eksplisiittisesti yhtälöstä 2.10 muotoon

$$\left|\psi(t)\right\rangle = U\left|\psi(0)\right\rangle, \quad U = e^{\frac{-iEt}{\hbar}}. \tag{2.10}$$

Stationääriselle systeemille ainoastaan vaihetekijä U on riippuvainen ajasta. Systeemiä kutsutaan stationääriseksi siksi, että sen todennäköisyystiheys pysyy aikaevoluutiossa vakiona.

2.4 Dekoherenssi

Kvanttimekaniikan interferenssi-ilmiöt ovat mahdollisia ainoastaan, jos vuorovaikuttavat systeemit ovat koherentteja keskenään. Systeemin vuorovaikuttaessa ympäsristönsä kanssa koherenssi vähenee, mitä kutsutaan dekoherenssiksi. Kun systeemi lomittuu ympäristönsä kanssa se ei ole enää esitettävissä tila-avaruudessaan yksikäsitteisenä aaltofunktiona, vaan aaltofunktioiden jakaumaa kuvaavana tiheysmatriisina. [3, s.58-63]

Jos systeemi olisi täysin ympäristöstään eristetty pysyisi se ikuisesti koherenttina. Käytännössä systeemillä esiintyy kuitenkin aina vuorovaikutusta ympäristönsä kanssa. Aikaa, jossa systeemissä on havaittavissa vaihekorrellaatiota kutsutaan koherenssiajaksi. Dekoherenssin seurauksena järjestelmän kvantti-ilmiöt häviävät ja superpositiotilat romahtavat klassisiksi tiloiksi. Dekoherenssi on syy sille, että makroskooppisten kvanttisysteemien havaitseminen on vaikeaa. [3, s.58-63]

3. KVANTTI-INFORMAATIO

Tietoa, joka sisältyy kvanttimekaanisen systeemin tilaan kutsutaan kvantti-informaatioksi. Se eroaa huomatttavasti klassisen systeemin sisältämästä informaatiosta.

3.1 Kubitti

Klassinen laskenta perustuu biteillä suoritettaviin operaatioihin. Bitti on yksikkö, jolla on kaksi mahdollista tilaa, 0 ja 1. Fyysisesti bitin tiloja vastaa usein tietty sähkövirta tai jännite tietokoneen komponentissa. Klassisen laskennan bittiä vastaava kvanttilaskennan perusyksikkö on kvanttibitti eli kubitti. Kubitti on kvanttimekaaninen systeemi, jolla on kaksi bitin tiloja 0 ja 1 vastaavaa ominaistilaa $|0\rangle$ ja $|1\rangle$. Tietyt mikroskooppiset systeemit kuten atomi, ytimen spin tai polarisoitu fotoni voivat toimia kubitteina [2]. Kubittien fyysiseen toteutukseen palataan tarkemmin luvussa 6. Klassisesta bitistä poiketen kubitilla on lukematon määrä mahdollisia tiloja, sillä se voi esiintyä ominaistilallaan tai millä tahansa näiden superpositioista.

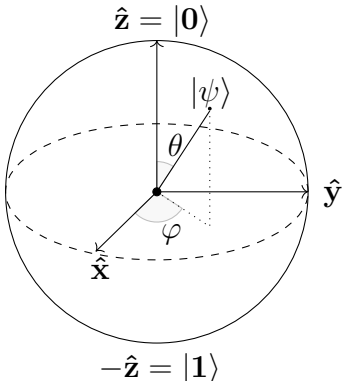
Kubitin tila $|\psi\rangle$ voidaan esittää yhtälön 2.5 mukaisesti superpositiona ominaistilojen virittämässä Hilbertin avaruudessa bra-ket-notaatiolla

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{3.1}$$

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle\,. \tag{3.2}$$

Kubitin tila-avaruuden geometrinen tulkinta tunnetaan Blochin pallona [9], joka on esitetty kuvassa 3.1. Blochin pallossa kubitin ominaistilat ovat pallon ja z-akselin leikkauspisteissä. Myös klassinen bitti voidaan esittää Blochin pallossa, mutta sen sallitut tilat rajoittuvat vain näihin kahteen pisteeseen. Kubitille sen sijaan kaikki pisteet pallon pinnalta ovat mahdollisia.

Blochin pallossa todennäköisyysamplitudit kubitin superpositiotilassa 3.2 saavat muodon



Kuva 3.1. *Blochin pallo*

$$\alpha = \cos\left(\frac{\Theta}{2}\right) \tag{3.3}$$

$$\beta = e^{i\phi}\sin\left(\frac{\Theta}{2}\right), \tag{3.4}$$

jossa $e^{i\phi}$ on kubitin aaltofunktion suhteellinen vaihe. Vaiheella ei ole vaikutusta kubitin todennäköisyysamplitudien suuruuteen. Kulma Θ siis määrittää ominaistiloja vastaavat todennäköisyydet. Kubitit ovat yleisesti stationäärejä systeemejä, jolloin todennäköisyystiheys ei ole riippuvainen aikaevoluutiosta.

Kubitti voi olla esimerksiksi niin kutsutussa Bellin tilassa $|\psi_B\rangle$, joka vastaa täydellistä superopsitiota

$$|\psi_B\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle + |1\rangle\right). \tag{3.5}$$

Tullessaan mitatuksi kubitilla on yhtäsuuri mahdollisuus romahtaa tilaan $|0\rangle$ kuin $|1\rangle$. Kuten luvussa 2 todettiin, on kvanttimekaanisen systeemin kuten kubitin tarkan tilan määrittäminen mittamalla mahdotonta. Kun kubitin tila mitataan, saadaan tulokseksi vain, jokin mitatavan suureen ominaisarvoista. Saadusta tuloksesta ei voi päätellä oliko kubitti superpositiossa ennen mittausta. Kvanttimekaniikan luonteesta johtuen, kvanttilaskennassa käsitellään todennäköisyyksiä, eikä laskenta ole klassisen laskennan tavoin deterministista.

3.2 Usean kubitin systeemi

Kvanttilaskennassa kvantti-informaation tallentamiseen ja käsittelyyn käytetään yleisesti usean kubitin systeemeitä. Kahden kubitin $|a\rangle$ ja $|b\rangle$ yhdistetty tila voidaan muodostaa tensoritulona, jolloin niiden muodostaman syseemin tila voidaan esittää yksikäsitteisesti

vektorina [3, p. 50]

$$|ab\rangle = |a\rangle \otimes |b\rangle = v_{00}\,|00\rangle + v_{01}\,|01\rangle + v_{10}\,|10\rangle + v_{11}\,|11\rangle \rightarrow \begin{bmatrix} v_{00} \\ v_{01} \\ v_{10} \\ v_{11} \end{bmatrix}. \tag{3.6}$$

Kahden kubitin systeemin tilan tarkkaan kuvaukseen vaaditaan siis neljä kompleksilukua. Kahden bitin tilan kuvaaminen sen sijaan vaatii ainoastaan kaksi binäärilukua. Yleisemmin, klassisella systeemillä, joka sisälttää n bittiä on 2^n erilaista tilaa, tilasta $x = 000\ldots0$ tilaan $x = 111\ldots1$. Vastaavasti n:n kubitin systeemi voi olla, millä tahansa tilalla Ψ, jotka ovat muotoa [2]

$$|\Psi\rangle = \sum_{00\ldots0}^{11\ldots1} \psi_x\,|x\rangle\,, \tag{3.7}$$

jossa ψ_x ovat 2^n-suuruinen joukko kompleksilukuja, jotka toteuttavat normeerausehdon $\sum_x |\psi_x|^2 = 1$. Jokainen systeemin tila on siis kompleksinen yksikkövektori 2^n-dimensioisessa Hilbertin avaruudessa

$$|\Psi\rangle \in \mathcal{H}^{2^n} = (\mathcal{H}_2)^n. \tag{3.8}$$

Tila-avaruuden eksponentiaalisesti kasvava ulotteisuus erottaa kvanttitietokoneet klassisista tietokoneista, joiden tila on aina kuvattavissa lineaarisesti kasvavalla määrällä parametrejä. Klassisen systeemin tila voidaan aina esittää kuvaamalla jokaisen sen osan tila erikseen. Suuri osa kvanttimekaanisen systeemin tiloista ovat lomittuneita, eikä vastaava kuvaus ole mahdollinen. Juuri lomittuneiden tilojen säilyttäminen ja käsittely ovat kvanttitietokoneiden laskennallisen tehon salaisuus ja toisaalta myös yksi niiden toteuttamisen suurimmista haasteista. [2]

3.3 Kvanttilaskennan mallit

Kvanttilaskennan prosessille on kehitetty monia erilaisia malleja. Merkittävinä niistä voi pitää adiabaattista, topologista ja mittaukseen perustuvaa kvanttilaskentaa sekä Turingin kvanttitietokonetta ja kvanttipiirimallia.

Turingin kone on Alan Turingin esittelemä laskennallinen malli, joka tarjoaa yksinkertaisen universaalin matemaattisen esityksen tietokoneen toiminnalle [10]. Kaikki klassiset algoritmit ovat esitettävissä Turingin koneen avulla, eikä yksikään klassinen tietokone voi olla

tehokkaampi kuin Turingin kone. Se antaa teoreettisen ylärajan tietokoneen laskennalliselle tehokkuudelle, joten sitä pidetään laskennallisen ongelman kompleksisuuden mittarina. Turingin kvanttitietokone on mallin laajennos, jonka avulla myös kvanttilaskenta on esitettävissä. Turingin kvanttitietokone on vain abstrakti laite, eikä sille ole fyysistä toteutusta. Sen avulla on kuitenkin määriteltävissä, mikä on kvanttilaskennallisesti mahdollista. [11]

Adiabaattinen kvanttilaskenta on malli, joka perustuu systeemin Hamiltonin operaattorin hyödyntämiseen. Hamiltonin operaattorin matemaattinen muoto on kullekin tilalle ominainen, ja sen muoto vaihtelee suuresti. Tämän vuoksi on mahdollista löytää Hamiltonin operaattori, joka vastaa haluttua laskennallista ongelmaa ja sen perustila ongelman ratkaisua. Laskenta etenee niin, että aluksi systeemi, jolla on yksinkertainen Hamiltonin operaattori, valmistellaan perustilalleen. Alkutilasta systeemin tilaa modifioidaan kohti monimutkaisempaa lopputilaa. Kun systeemille tehtävät muutokset suoritetaan riittävän hitaasti, pysyy systeemi perustilallaan, jolloin lopputilan Hamiltonin operaattori sisältää ongelman ratkaisun. [12] Eräs adiabaattisen kvanttilaskentaan perustuva menetelmä on niin kutsuttu kvanttijäähdytys, jota voidaan käyttää monimutkaisten optimointiongelmien ratkaisemisessa [13].

Topologinen kvanttilaskenta on Aleksei Kitaevin esittelemä kvanttilaskennan malli, jossa informaatio tallennetaan anyoneiksi kutsuttuihin kvasihiukkasiin. Laskenta tapahtuu siirtelemällä anyoneita toistensa suhteen kaksidimensioisessa avaruudessa, jolloin niiden maailmanviivat risteävät kolmedimensioisessa aika-avaruudessa. Solmukohdat aika-avaruudessa muuttavat anyoneiden fyysistä tilaa. Siirtelemällä anyoneita tietyssä järjestyksessä voidaan näin suorittaa loogisia operaatioita kohti haluttua lopputulosta. [14]

Mittaukseen perustuvassa kvanttilaskennassa kaikki lähtöinformaatio tallennetaan aluksi monen kubitin lomittuneeseen tilaan. Laskenta tapahtuu mittaamalla sen jälkeen yksittäisiä kubitteja tietyssä järjestyksessä. Kun yksittäinen kubitti mitataan, vaikuttaa se suoraan muuhun lomittuneeseen systeemiin. Mittaamalla järjestelmällisesti yksittäisiä kubitteja voidaan dataa prosessoida halutulla tavalla. [15] Kubitin mittaaminen ei ole reversiibeli prosessi vaan aiheuttaa osan informaation tuhoutumisen, minkä takia kyseisellä menetelmällä toimivia kvanttitietokoneita kutsutaan myös yksisuuntaisiksi kvanttitietokoneiksi [16].

Kvanttipiirimallissa informaatio on tallennettu kubitteihin, joihin kohdistetaan sarja kvanttiportteja lopputuloksen saamiseksi. Kvanttiportit ovat yhdelle tai muutamalle kubitille suoritettavia loogisia operaatioita. Kvanttipiirit on malleista lähinnä klassisen digitaalisen tietokoneen toimintaa. Mallissa kubitit vastaavat klassisia bittejä ja kvanttiportit klassisen laskentapiirin logiikkaportteja. Suuri osa tunnetuista kvanttialgoritmeista on suunniteltu kvanttipiirimallilla. [13]

Edellä esitetyt mallit ovat laskennallisesti ekvivalentteja keskenään [12] [16] [17] [18]. Toi-

sin sanoen mikä tahansa algoritmi voidaan muokata mallista toiseen ilman merkittävää tehokkuuden menetystä. Huomion arvoista on se, että mallit ovat ekvivalentteja Turingin koneen kanssa [18], eli edellä mainitut mallit ovat laskennallisesti niin tehokkaita, kuin teoreettisesti on mahdollista [11].

4. KVANTTIPIIRIT

Kvanttipiirit on vallalla oleva kvanttilaskennanmalli, jonka David Deutsch esitteli 1989 saattaakseen kvanttilaskennan fysikaalisesti helposti tulkittavaan muotoon [19, s. 235]. Mallissa kvanttilaskenta kuvataan kvanttilogiikkaporttien kytkentöjen muodostamina piireinä tai verkkoina, jotka muistuttavat klassisia logiikkapiirejä. Kvanttipiirin kytkennät ovat vertauskuvallisia, sillä piirin muodostavat kubittijoukkoon peräkkäin kohdistettavat kvanttioperaatiot.

4.1 Kvanttiportit

Kvanttipiirimallissa kvanttilogiikkaportti, lyhyemmin kvanttiportti, on kvanttilaskennan pienin askel. Laskennallisesti kvanttiportti on pieneen määrään kubitteja operoiva unitaarimuunnos. Unitaarioperaattorille U pätee $U^\dagger = U^{-1}$. Mikä tahansa unitaarimuunnos on matemaattisesti rakennettavissa kahteen kubittiin kerrallaan kohdistuvista unitaarimuunnoksista. [2] Tämän vuoksi universaalia kvanttilaskentaa varten riittää ainoastaan yhden ja kahden kubitin kvanttiporttien määrittely. Laskennan lopputulokseen päästään yhdistelemällä kvanttiportteja tietyssä järjestyksessä tietyille kubiteile.

Siinä missä klassisia yhden bitin logiikkaportteja on vain yksi, NOT-portti, on mahdollisia yhden kubitin portteja lukematon määrä [2]. Kaikista yleisimmässä tapauksessa yhden kubitin portti on 2×2 unitaarinen matriisioperaattori [2]

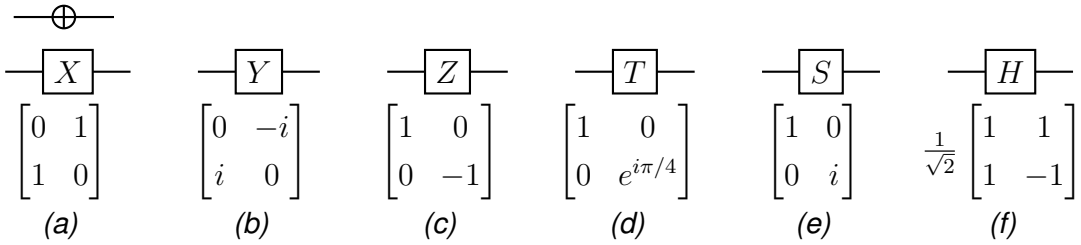
$$U = \begin{bmatrix} \alpha & \gamma \\ \beta & \delta \end{bmatrix}, \quad \alpha, \gamma, \beta, \delta \in \mathbb{C} \tag{4.1}$$

jonka operoidessa kubitin tiloihin $|0\rangle$ ja $|1\rangle$ saadaan tilat

$$U|0\rangle = \alpha|0\rangle + \beta|1\rangle, \quad U|1\rangle = \gamma|0\rangle + \delta|1\rangle. \tag{4.2}$$

Kubitin saattamiseksi täysin mielivaltaisesti mihin tahansa tilaan vaadittaisiin ääretön määrä erilaisia kvanttiportteja. Fyysisen laskennan kannalta tämä on kuitenkin mahdotonta, sillä käytettävien perusoperaatioiden on määrä on oltava äärellinen. Se ei kuitenkaan ole laskennan kannalta ongelma, sillä mikä tahansa unitaarinen operaatio on approksimoita-

vissa riittävällä tarkkuudella käyttäen yhdistelmää äärellisestä määräästä kvanttiportteja. [20] Käytännössä universaaliin laskentaan tarvittavien kvanttiporttien määrä voidaan rajata pieneksi [2].

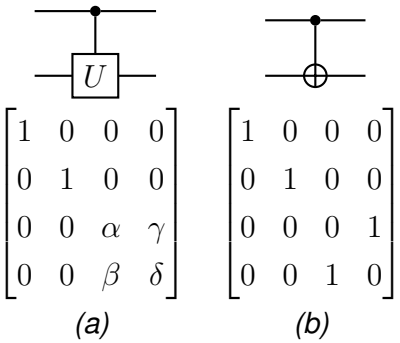


Taulukko 4.1. *Yleiset yhden kubitin kvanttiportit*

Taulukossa 4.1 on esitetty yleisesti käytettyjä yhden kubitin kvanttiportteja ja niitä vastaavat matriisioperaattorit. Koska porttioperaatiot ovat unitaarisia, voidaan yhden kubitin kvanttiportit esittää Blochin pallossa vektorin kiertämisenä operaattorille ominaisen akselin ympäri. Vektorin kierto π radiaanin verran eri kordinaattiakselien ympäri on yleinen operaatio, jolle on nimetty kordinaattiakseleiden mukaan portit Pauli-X, -Y ja -Z 4.1*(a)(b)(c)*. Näistä Pauli-X-portti on kvanttipiirin vastine klassiselle NOT-portille. Se muuttaa ominaistilan $|0\rangle$ tilaksi $|1\rangle$ ja toisin päin. Pauli-Y muuttaa ominaistilan $|0\rangle$ tilaksi $i\,|1\rangle$ ja ominaistilan $|1\rangle$ tilaksi $i\,|0\rangle$.[2]

Yleisesti Z-akselin kiertoa kuvaavia portteja kutsutaan vaiheensiirtoporteiksi. Niiden avulla voidaan muuttaa kubitin vaihetta vaikuttamatta kubitin todennäköisyyteen tulla mitatuksi tilana $|0\rangle$ tai $|1\rangle$. Usein käytettyjä vaiheensiirtoportteja ovat kulmaa π vastaava Pauli-Z-portti, kulmaa $\frac{\pi}{4}$ vastaava T-portti 4.1*(e)* ja kulmaa$\frac{\pi}{2}$ vastaava S-portti 4.1*(e)*.

Näiden lisäksi kvanttilaskennan kannalta erityisen tärkeä operaatio on kubitin saattaminen ominaistilalta täydelliseen superpositioon. Tämä voidaan toteuttaa Hadamardin muunnoksella, jota kvanttipiireissä vastaa Hadamardin portti 4.1 *(f)*. [21] Hadamardin muunnos kuvautuu Blochin pallossa kierroksi π radiaania vektorin $(\hat{x} + \hat{z})/\sqrt{2}$ ympäri.



Taulukko 4.2. *(a) Kontrolloitu yleinen portti U ja (b) kontrolloitu Pauli-X-portti, CNOT-portti.*

Lomittuneen tilan luomiseksi käytetään kvanttipiireissä yleisesti yhden kubitin portteja, joita kontrolloidaan toisella kubitilla [13]. Kontrolloitu portti ottaa syötteenään kaksi kubit-

tia, joista toinen on kontrolli- ja toinen maalikubitti. Portti muuttaa maalikubitin tilaa riippuen kontrollikubitin arvosta. Esimerkiksi yleisesti käytetty kontrolloitu Pauli-X-portti 4.2 eli CNOT-portti, vaihtaa maalikubitin arvon kontrollikubitin arvon ollessa $|1\rangle$. Kontrollikubitin arvon ollessa $|0\rangle$ pysyy maalikubitin arvo muuttumattomana. [2] [13]. Kahden kubitin portit voidaan yleisesti kuvata 4×4 matriiseina, jotka operoivat vektoriin 3.6. Toisella kubitilla kontrolloitu yleinen portti 4.1 ja CNOT-portti on esitetty taulukossa 4.2.

Yhden ja kahden kubitin porteista voidaan muodostaa erilaisia universaaleja ryhmiä [13]. Esimerkiksi porttien CNOT, H, S ja T avulla voidaan suorittaa tehokkaasti mikä tahansa kvanttitietokonella mahdollinen laskutoimitus [2]. Käytettävät portit määrätytyvät kvanttitietokoneen fyysisen toteutuksen perusteella.

Toisin kuin yleisesti käytetyt klassisest logiikkaportit, kvanttiportit ovat täysin reversiibeleitä [11]. Jokainen systeemiin kohdistettu porttioperaatio U voidaan kumota kohdistamalla systeemiin sen käänteismuunnos $U^{-1} = U^{\dagger}$. Reversiibelissä laskentaprosessissa ei synny entropiaa eikä lämpöä.

4.2 Deutschin ongelma

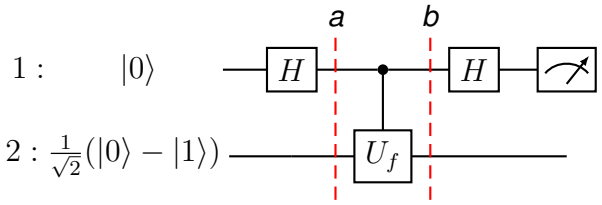
David Deutsch esitteli 1985 kanttialgoritmin, joka ensimmäistä kertaa osoitti kvanttilaskennan tehokkuuden. Algoritmi liittyy matemaattisen funktion $f(x)$ laskemiseen. [21] Määritellään $f(x)$ on binääriseksi funktioksi, joka kuvaa määrittelyjoukon $\{0, 1\}$ alkion arvojoukkoon $\{0, 1\}$. Mahdollisia funktioita on neljä

$$(a)\begin{cases} f(0) = 0 \\ f(1) = 0 \end{cases} \quad (b)\begin{cases} f(0) = 1 \\ f(1) = 1 \end{cases} \quad (c)\begin{cases} f(0) = 0 \\ f(1) = 1 \end{cases} \quad (d)\begin{cases} f(0) = 1 \\ f(1) = 0 \end{cases} \tag{4.3}$$

Funkiot voidaan jakaa vakiofunktioihin ja tasapainotettuihin sen mukaan saako funktio saman arvon muuttujan x eri arvoilla. Deutschin esittämä laskennallinen ongelma oli: Onko funktio f tasapainotettu? [21]

Klassisella tietokoneella funktio on laskettava kahdesti tuloksen saamiseksi. Ensin molemmilla arvoilla ja vastaus saadaan vertaamalla saatuja tuloksia keskenään. Deutsch osoitti, että kvanttitietokoneella ongelma voidaan kuitenkin ratkaista laskemalla funktio ainoastaan yhden kerran. Kvanttitietokoneella voidaan siis ongelman ratkaisussa saavuttaa kaksinkertainen nopeus verrattuna klassiseen tietokoneeseen. Deutschin algoritmin onnistumisen todennäköisyys oli vain 50 $\%$. Myöhemmin ongelman ratkaisemiseksi on kehitetty algoritmi, joka antaa vastauksen 100 $\%$:n varmuudella [21].

Paranneltu Deutschin algoritmi on esitetty kahden kubitin kvanttipiirinä kuvassa 4.1. Kvanttipiiriä luetaan vasemmalta oikealle. Kubitin 1 syötearvona on $|0\rangle$ ja kubitin 2 superposi-



Kuva 4.1. *Paranneltu Deutschin algoritmi. Portti U_f on $f(x)$-kontrolloitu NOT-portti.*

tio $\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$. Aluksi kubitille 1 suoritetaan Hadamardin muunnos, jonka jälkeen se on $1/\sqrt{2}(|0\rangle + |1\rangle)$. Seuraavaksi kubitin 1 arvolla säädetään funktiolla $f(x)$ kontrolloitua NOT-porttia U_f, jonka toiminta voidaan matemaattisesti muotoilla kaavalla [21]

$$|x\rangle\,|y\rangle \xrightarrow{f-c-N} |x\rangle\,|y \oplus f(x)\rangle\,, \tag{4.4}$$

jossa $|x\rangle$ on kontrollikubitin ja $|y\rangle$ syötekubitin tila. Operaatio $\oplus$ on yhteenlasku $mod(2)$. Systeemin tilan muutos tilasta *a* tilaan *b* saa muodon

$$\frac{1}{2}\underbrace{(|0\rangle + |1\rangle)}_{1}\underbrace{(|0\rangle - |1\rangle)}_{2} \longrightarrow \frac{1}{2}\underbrace{(-1)^{f(0)}\left(|0\rangle + (-1)^{f(0)\oplus}\,|1\rangle\right)}_{1}\underbrace{(|0\rangle - |1\rangle)}_{2} \tag{4.5}$$

Kubitit ovat kohdassa *b* lomittuneet ja sisältävät superpositioina molemmat lähtöarvot (kubitti 1) sekä niitä vastaavat funktion arvot (kubitti 2). Vastauksen saamiseksi on suoritettava mittaus. Jos kubitit mitataan kohdassa *b*, superpositio romahtaa ja saadaan vastauksena ainoastaan toista syötettä vastaava tulos. Tilanne ei ole tällöin yhtään klassista tietokonetta parempi. Vastaus voidaan kuitenkin määrittää hyödyntmällä superpositiotilojen interferenssiä. [21]

Jätetään kubitti 2 mittaamatta ja siirrytään tarkastelemaan ainoastaan kubitin 1 tilaa kohdassa *b*. Huomataan, että se saa erilaisia arvoja riippuen funktiosta f, jotka ovat

$$(a)\ \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$$
$$(b)\ \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$$
$$(c)\ \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$$
$$(d)\ \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle).$$

Suoritetaan kubitille 1 vielä yksi Hadamardin muunnos. Funktion f ollessa vakiofunktio

((a) ja (b)) kiertyy kubitti ominaistilalleen $|0\rangle$ ja funktion ollessa tasapainotettu kiertyy kubitti tilaan $|1\rangle$. Koska kubitti on nyt ominaistilallaan, saadaan mittauksella ongelmaan yksikäsitteinen ratkaisu. [21]

Vaikka Deutschin algoritmi on yksinkertainen, on se hyvä kuvaus, kuinka kvanttilaskentaa voidaan yleisesti hyödyntää ongelman ratkaisussa. Monet Deutschin algoritmin piirteet ovat kvanttialgoritmeille tunnusomaisia [21]. Algoritmi saavuttaa klassista laskentaa suuremman nopeuden kvanttirinnakkaisuuden avulla. Kvanttirinnakkaisuudeksi kutsutaan kvanttitietokoneen kykyä käsitellä useita klassisia syötearvoja samanaikaisesti superpositiotilojen avulla [11]. Toinen kvanttialgoritmeille yleinen piirre on myös se, että tulos on muotoiltava superpositiotilasta klassiseksi tilaksi yksikäsitteisen tuloksen saamiseksi [21].

5. KVANTTIALGORITMIT

Klassinen algoritmi on yksityiskohtainen kuvaus, kuinka jokin ongelma voidaan ratkaista vaihe vaiheelta. Kvanttialgoritmiksi kutsutaan algoritmeja, jotka hyödyntävät kvanttiilmiöitä, kvanttisuperpositiota tai lomittumista.

5.1 Kvanttialgoritmien tehokkuus

On syytä uskoa, että kvanttitietokone voi suoriutua, joistain tehtäväistä paremmin kuin yksikään klassinen tietokone [22]. Tilanteessa, jossa kvanttitietokone suoriutuu tehtävästä, joka olisi klassisella tietokoneella mahdoton, sanotaan kvanttitietokoneen saavuttaneen kvanttiherruuden [23]. Nämä määritelmät ovat kuitenkin sellaisenaan epämääräisiä. On mahdollista tarkentaa, kuinka suuri parannus on, ja mikä on klassisella tietokoneella mahdollista.

Laskenta on kaikissa tilanteissa fyysinen prosessi. Tämän vuoksi rajoittavina tekijöinä ovat aina tila (muisti) ja aika. Laskennassa saavutettava parannus tarkoittaa, että laskennallisen ongelman suorittaminen onnistuu, joko pienemmässä tilassa tai lyhyemmässä ajassa. [20]

Kompleksisuuden teoriana tunnetulla tieteenalalla tutkitaan, kuinka paljon ongelman ratkaisu vaatii aikaa ja tilaa sen syötekoon kasvaessa. Kahden ongelman ratkaisevan algoritmin sanotaan olevan laskennallisesti ekvivalentteja keskenään, jos toisen ratkaisun nopeus t_2 on esitettävissä ensimmäisen ratkaisun kestosta t_1 riippuvana polynomifunktiona f [12]

$$t_2(N) \propto f(t_1(N)) \tag{5.1}$$

syötekoon N kasvaessa. Yleisesti kahden algoritmin suorituskykyä vertailtaessa huomio kiinnittyy funktion $f(t)$ muotoon.

Algoritmien luokitteluun suoritusajan kasvun perusteella käytetään usein O-notaatiotiota. Notaatiossa käytetään merkintää $O(g(N))$, jossa funktio $g(N)$ on suoritusaikaa aidosti ylhäältä rajoittava syötekoosta N riippuva funktio. Toisin sanottuna, jos algoritmin suoritusaika on $O(g(N))$, ei algoritmin suoritusaika t huonoimmassakaan tapauksessa ylitä

riippuvuutta $t \propto g(N)$ syötekoon N kasvaessa.

Ongelmien kompleksisuutta käsiteltäessä voidaan tehdä karkea jako käsiteltäviin ja ei-käsiteltäviin ongelmiin. Ongelma on käsiteltävä, jos se on mahdollista ratkaista syötekoosta riippuen polynomimuotoisessa ajassa. Klassisella tietokoneella polynomisessa ajassa ratkaistavien ongelmien joukko on $\mathcal{P}$. Joukon $\mathcal{P}$ ongelmien lisäksi on suuri joukko ongelmia, joiden ratkaisemiseksi ei ole löydetty polynomiaalista algoritmia vaan esimerkiksi eksponentiaalisia $O(\alpha^N)$. Tälläisien ongelmien ratkaisemisen katsotaan olevan niin hidasta, että niitä pidetään kompleksisuuden teoriassa ei-käsiteltävinä. [4] [13]

Tällä hetkellä pidetään hyvin todennäköisenä, että kvanttitietokoneella polynomisessa ajassa ratkaistavien ongelmien joukko $\mathcal{BQP}$ on suurempi kuin $\mathcal{P}$. Tämä tarkoittaisi sitä, että kvanttilaskennalla olisi mahdollisuus muotoilla yleisesti ei-käsiteltäviksi luokiteltuja ongelmia käsiteltäviksi. Tätä ei ole kuitenkaan toistaiseksi pystytty todistamaan matemaattisesti. [13]

Tällä hetkellä tunnetaan useita algoritmeja, jotka ovat tunnettuja klassisia algoritmeja nopeampia. Edellä esitetty Deutschin algoritmi on hyvä esimerkki. Erityisen kiinnostavia ovat algoritmit, jotka tuottavat eksponentiaalisen nopeuden lisäyksen. Tälllaisia algoritmeja tunnetaan toistaiseksi vain muutamia [24]. Löydetyistä kvanttialgoritmeista merkittävimpinä voidaan pitää Shorin ja Groverin algoritmeja sekä kvanttisimulaatiota ja HHL-algoritmia.

5.2 Shorin algoritmi

Kokonaisluvun alkutekijöihin jako, on vaikea matemaattinen ongelma. Suurille luvuille se on niin vaikeaa, että monet nykyisin laajasti käytetyt kryptografiset menetelmät, kuten RSA-salaus, pohjautuvat ongelman vaikeuteen [25]. Vuonna 1994 Peter Shor esitti ongelman ratkaiseva algoritmin [4], joka on todennäköisesti kaikista tunnnetuin kvanttialgoritmi. Tehokkaimmankin tunnetun klassisen algoritmin suoritusaika t kasvaa lähes eksponentiaalisesti kokonaisluvun pituuden N kasvaessa. Shorin algoritmi sen sijaan hidastuu ainoastaan polynomiaalisesti. Kvanttilaskennalla voitaisiin siis saavuttaa ongelmassa lähes eksponentiaalinen nopeuden kasvu.

Shorin esittelemä algoritmi perustuu funktion

$$F_N(x) \equiv a^x \mod N \quad a \in [0, N], \tag{5.2}$$

käyttäytymiseen. Funktiossa esiintyvä parametri a on satunnaisesti valittava kokonaisluku. Tiedetään, että tarkastaltessa funktion käyttäytymistä x:n kasvaessa muodostavat funktion $F_N(x)$ arvot jaksollisen sarjan. Sarjan jaksonpituus r on kullekin luvulle N ominainen, eikä riippuvuudelle ole tunnettua kaavaa. Kun jakso löydetään saadaan luvun N

alkulukutekijät etsimällä lukujen N ja $a^{\frac{r}{2}} \pm 1$ suurin yhteinen tekijä. Suurimman yhteisen tekijän haku onnistuu tehokkaasti Eukeleideen algoritmilla.

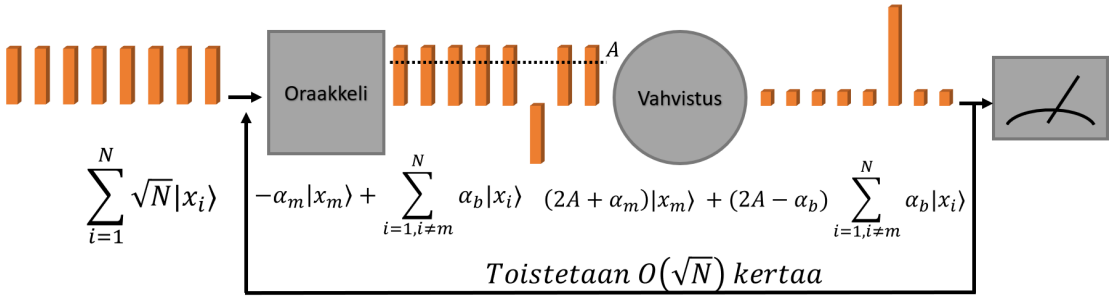
Shorin algoritmi vaatii kaksi kubittirekisteriä, joiden koko on $L \simeq N^2$. Ensimmäinen rekisterin jokainen kubitti valmistellaan aluksi täydelliseen superpositioon. Tämän jälkeen rekisteri sisältää täydellisen superposition, jokaisesta kokonaisluvusta nollan ja luvun $2^L - 1$ väliltä. Ensimmäistä rekisteriä käytetään syötteenä kvanttitietokoneelle, joka laskee funktion F_N arvon. Operaation jälkeen ensimmäisen rekisterin lähtöarvot sekä toisessa rekisterissä sijaitsevat funktion arvot ovat lomittuneet toisiinsa. Seuraavaksi toisen rekisterin kubitit mitataan ja vastaukseksi saadaan, jokin funktion F_N arvo. Tämän arvo ei varsinaisesti ole kiinnostava. Mittaus suoritetaan, sillä se romahduttaa toisen rekisterin arvon, jolloin ensimmäiseen rekisteriin jää ainoastaan tätä funktion arvoa vastaavat lähtöarvot. Ensimmäinen rekisteri sisältää nyt lähtöarvot. Varsinaisesten lähtöarvojen sijasta kiinnostavaa on niiden jaksollisuus.

Jaksollisuuden selvittämiseksi on Fourier'n muunnos tunnetusti hyvä työkalu. Fourier'n muunnokselle tunnetaan tehokas yksinkertainen kvanttialgoritmi [26], joka on monen monimutkaisemman tehokkaan kvanttialgoritmin perusta. Kun ensimmäiselle rekisterille suoritetaan nyt Fourier'n muunnos saadaan mitattua jakso r suurella todennäköisyydellä. Toistamalla operaatio $O(\log N)$ kertaa lähestyy todennäköisyys yhtä. Shorin algoritmin nopeus saavutetaan suurella kvanttirinnakkaisuudella, tilojen interfernessi-ilmiöillä sekä rekistereiden tilojen lomittumisella, kuten myös aiemmin esitetyn Deutschin algoritminkin tapauksessa.

5.3 Groverin algoritmi

Järjestelemättömästä tietokannasta tiedon hakeminen on tietojenkäsittelytieteissä yleinen laskennallinen ongelma. Lov Grover esitti vuonna 1996 kvanttihakualgoritmin [27], joka suoriutuu tästä tehtävästä merkittävästi klassisia algoritmeja tehokkaammin. Algoritmi tunnetaan Groverin algoritmina. Grover käytti artikkelissaan esimerkkinä järjestelemättömästä tietokannsta elektronista puhelinluetteloa, johon tehdään hakuja puhelinnumeron perusteella, sen ollessa järjestetty nimien mukaan aakkosjärjestykseen. Kun haku tehdään klassisilla algoritmeilla on käytävä jokainen puhelinnumero yksitellen läpi, kunnes löydetään etsitty nimi. Hakutuloksen saamiseksi on keskimäärin tarkistettava puolet puhelinnumeroista, $N/2$ kappaletta ja huonoimmassaa tapauksessa kaikki. Klassisilla algoritmeilla voidaan saavuttaa ongelmassa siis lineaarinen tehokkuus $O(N)$. Grover osoitti, että kvanttitietokone voisi suoritua tehtävästä ajassa $O(\sqrt{N})$. [27]

Groverin algoritmissa on neljä vaihetta, jotka ovat alustaminen, oraakkeli, vahvistus ja mittaaminen. Aluksi kubittirekisteriin alustetaan kaikkien hakuavainten yhtäläinen superpositio, jossa kaikkien termien amplitudi on $1/\sqrt{N}$. Algoritmi vaatii toimiakseen kvanttipiirin U_f, joka laskee boolisen funktion $f(x)$. Funktio palauttaa hakukriteerit täyttävälle alkiol-



Kuva 5.1. *Superpositiotilan termien kehitys Groverin algoritmin eri vaiheissa. Kuvassa $|x_b\rangle$ on etsitty arvo ja A on termien amplitudien keskiarvo oraakkelipiirin jälkeen. (mukaillen [28])*

le arvoksi 1 ja muutoin 0. Kvanttipiiriä U_f kutsutaan oraakkeliksi. Kun oraakkelipiiri ajetaan alustetulle kubittirekisterille saadaan tulokseksi funktion $f(x)$ arvojen superpositio. Oraakkelin tuottaman superpositiotilan ja hakuavainten superpositiotilojen interferenssi-ilmiöiden avulla on nyt mahdollista suorittaa niin sanottu diffuusio-operaatio. Operaatio kasvattaa superpositiotilan termiä, joka vastaa funktion $f(x)$ arvoa 1 ja pienentää muiden amplitudia. Etsityn termin amplitudi saavuttaa maksimin, kun diffuusio-operaatio on toistettu $\pi/4\sqrt{N}$ kertaa. Tällöin termi selvästi dominoi superpositiotilaa. Kun rekisteri lopuksi mitataan, saadaan suurella todennäköisyydellä etsitty arvo.[27]

Vaikka Grover esitteli algoritmin alunperin hakualgorimtina, jolla pystytään tekemään hakuja tietokannasta, ei sen käyttö kirjaimellisesti tietokantoihin tehtäviin etsintöihin ole realistista. Groverin algoritmin ajamiseksi on tiedon oltava tallennettu kubittien superpositiotiloihin, joka pitäaikaisessa tallennuksessa on hyvin herkkä dekoherenssin aiheuttamalle tiedon menettämiselle. Sen sijaan Groverin algoritmi voisi haastaa klassiset algoritmit niin kutsuttuihin virtuaalisiin implisiittisiin tietokantoihin tehtävissä hauissa, kuten esimerkiksi kryptografisen avaimen etsinnässä. Monissa sovelluksissa Groverin algoritmin haasteena on oraakkelina toimivan kvanttipiirin U_f toteuttaminen. Jos ongelmalle ei löydy yksinkertaista oraakkelia, voi oraakkelin toteutus mitätöidä Groverin algoritmin mahdollistaman nopeudenlisäyksen. [29]

Grover on myös esittänyt, että algoritmia voidaan käyttää myös lukujoukon keskiarvon tai mediaanin laskemisessa [30]. Yleistetty Groverin algoritmi, amplitudin vahvistaminen [31], on toiminut pohjana monelle myöhemmin löydetylle kvanttialgoritmille.

5.4 HHL-algoritmi

Hyvin monet systeemit ovat insinööri- ja luonnontieteissä mallinnettavissa lineaarisen yhtälöryhmän avulla. Usein käsiteltävät systeemit ovat hyvin suuria, mikä tekee laskennasta hyvin raskasta. Vuonna 2009 Aram Harrow, Avinatan Hassidim ja Seth Lloyd esittelivät kvanttialgoritmin lineaaristen yhtälöryhmien ratkaisemiseen [24], joka tunnetaan

HHL-algoritmina.

Lineaarinen yhtälöryhmä on esitettävissä matriisiyhtälönä, jossa A on $N \times N$ matriisi, $\vec{b}$ tunnettu vektori ja $\vec{x}$ tuntematon.

$$A\vec{x} = \vec{b} \quad A = N \times N \tag{5.3}$$

HHL-algoritmia käytettäessä matriisin A tulee olla hermiittinen matriisi. Se ei kuitenkaan estä algoritmin hyödyntämistä muissakin tapauksissa, sillä mistä tahansa $N \times N$ matriisista voidaan muodostaa hermiittinen $2N \times 2N$ matriisi. Lisäksi matriisin A häiriöalttiuden κ on oltava riittävän pieni, $\kappa^{-2}I \leq A^{\dagger}A \leq I$.

Hyödyntämällä jo aiemmin esitettyjä Hamiltonin operaattorin simulaatioalgoritmia ja kvantti-Fourierin muunnosta HHL-algoritmi ratkaisee yhtälöryhmän muodostamalla matriisin A käänteismatriisin A^{-1} ajassa $O(\log(N)\kappa^2/\epsilon)$, jossa $\epsilon > 0$ on tarkkuusparametri. Parhaat klassiset algoritmit suoriutuvat lineaarisen yhtälöryhmän ratkaisussa parhaimmillaankin lineaarisessa ajassa yhtälöiden lukumäärään nähden $O(N)$ puhumattakaan matriisin kääntämisestä, joka on usein huomattavasti raskaampi operaatio. HHL-algoritmi on siis eksponentiaalisesti klassisia algoritmeja nopeampi. Huomionarvoista on, että algoritmin laskeman ratkaisuvektorin $\vec{x}$ lukeminen on lineaarinen operaatio. Logaritmisella tehokkuudella voidaan kuitenkin selvittää mikä tahansa ratkaisuvektoriin ja matriisiin M liittyvä skalaari $\vec{x}^{\dagger}M\vec{x}$, joka usein on varsinaista ratkaisuvektoria kiinnostavampi. [24]

On myös esitetty, kuinka algoritmia voidaan hyödyntää lineaaristen differentiaaliyhtälösysteemin ratkaisemisessa [24] ja sovitteiden ratkaisussa pienimmän neliösumman menetelmällä [32].

5.5 Kvanttimekaaisten systeeemien simulaatio

Simulaatio on prosessi, jossa rakennetulla systeemillä jäljitellään toisen systeemin käytöstä. Kvanttimekaanisten systeemien simulointi on klassisilla tietokoneilla hyvin raskasta. [22] Kvanttimekaanisen systeemin, jolla on N muuttujaa, täydellinen kuvaus vaatii 2^N tilavektorin käsittelyä. Klassisella tietokoneella muistin, sekä ajan, kulutus kasvaa eksponentiaalisesti muuttujien määrän kasvaessa $O(2^{2^N})$. Kvanttitietokoneen muistin ja ajan kulutus on tehtävässä lineaarinen simuloitavan systeemin kokoon N ja simuloitavaan ajanjaksoon t_0 nähden $O(Nt_0)$.[33]

Systeemille suoritettavat operaatiot, sekä Schrödingerin yhtälön mukainen aikaevoluutio voidaan simuloida halutulla tarkkuudella kvanttitietokoneen porttioperaatioiden avulla. Kvanttitietokoneella myös ympäristön aiheuttaman dekoherenssin vaikutus on simuloitavissa. Simulaatio olisi tällöin mahdollista niin lokaaleille kuin avoimille systeemeille. [33]

5.6 Virheenkorjaus

Pienikin interaktio ympäristön kanssa aiheuttaa dekoherenssia ja siten myös virheitä laskennassa. Yhdellä kubitin virheet on jaettavissa kolmeen tyyppiin. Kubitin tilan merkki voi vaihtua, mikä vastaa Pauli-Z-operaatiota. Toisena kubitin ominaistilojen merkki voi vaihtua keskenään, mikä vastaa Pauli-X-operaatiota. Kolmas vaihtoehto on yhdistelmä aiemmista ja vastaa näin Pauli-Y operaatiota. [34] [23]

Virheenkorjausalgoritmien idea on tallentaa yhden kubitin kvantti-informaatio yhden kubitin sijasta useaan lomittuneeseen kubittiin [34]. Jos yhdessä kubitissa ilmenee laskennan aikana virhe ei se vaaranna koko laskennan tulosta. Klassisten bittien tapauksessa vastaava menetelmä on helppo toteuttaa. Esimerkiksi bitin arvon ollessa 1 voidaan se tallentaa kolmeen bittiin tilaksi 111. Kun näille biteille nyt suoritetaan sama operaatio, voidaan lopputuloksia vertailemalla huomata, jos yhden bitin arvo on virheellinen. Kun käsitelllään kvantti-informaatiota tilanne on kuitenkin monimutkaisempi. Kubitin vantti-informaatiota ei voida kopioida [2]. Myöskään vastaavaa vertailua ei voida suorittaa, sillä säilyttääkseen kubitin superpositiot ja lomittumisen kubittia ei voida mitata. Kubitin virhe on korjattava ilman tietoa siitä, mikä kubitin tila on.[34]

Virheenkorjausalgoritmien idean esitteli Shor kehitettyään algoritmin [35], jossa yhden loogisen kubitin tila tallennetaan yhdeksään fyysiseen kubittiin yksinkertaisella kvanttipiirillä. Koodauksen jälkeen kaikki fyysiset kubitit käyvät läpi laskentapiirin, joka loogiselle kubitille halutaan suorittaa. Laskennan jälkeen loogisen kubitin koodaus voidaan purkaa yhteen fyysiseen kubittiin suorittamalla operaatiot käänteisessä järjestyksessä. Jos yhdessä kubitissa on tapahtunut virhe, piiri havaitsee mihin kubittiin virhe on kohdistunut, sekä mikä virhe on kyseessä. Algoritmi korjaa loogisen kubitin arvon sen mukaisesti.

Vaadittavien kubittien lukumäärää voidaan vähentää yhdeksästä. Shorin virheenkorjauksen jälkeen on esitetty seitsemän ja viiden kubitin virheenkorjausalgoritmeja. On osoitettu, että viisi kubittia on pienin mahdollinen lukumäärä, jolla virheenkorjaus voidaan suorittaa. [34]

Virheenkorjausalgoritmien suurin haaste on, että käytettävien fyysisten kubittien lukumäärän kasvaessa kasvaa laskennan komlpeksisuus ja siten myös virheiden todennäköisyys. Myös virheenkorjausalgoritmien operaatiot itsessään ovat muiden kvanttioperaatioiden tapaan virhealttiita. Tämän takia toimiakseen virheenkorjausalgoritmin toteutuksen tulee vähentää laskennassa tapahtuvia virheitä enemmän, kuin se niitä aiheuttaa. Tämä ei ole kuitenkaan osoittautunut fyysisesti helposti toteutettavaksi. [22]

6. KVANTTITIETOKONEIDEN KEHITYS

Kvanttitietokone oli pitkään ainoastaan teoreettinen malli. Fyysisen kvanttilaskennan tutkimus on edennyt viime vuosikymmeninä yksittäisten kubittien kokeellisista toteutuksista useiden kubittien operaatioihin ja kokeellisiin muutamien kymmenien kubittien kvanttitietokoneisiin.

6.1 Kvanttitietokoneiden fyysinen toteutus

Kubitteja on toteutettu fyysisesti monin eri tavoin, joihin lukeutuvat fotonit, ioniloukut, suprajohtavat virtapiirit, yksittäiset atomit ja puolijohdekvanttipisteet [36]. Vaikka yksittäiseksi kubitiksi soveltuvia systeemeitä on lukuisia, ei näistä suuri osa sovellu kvanttilaskentaan. Toimiakseen fyysisen kvanttitietokoneen tulee täyttää David P. DiVencenzon muotoilemat viisi kriteeriä [37], jotka ovat:

1. Skaalautuva kubittijärjestelmä
2. Mahdollisuus alustaa kubittien tila luotettavasti yksinkertaiseen tilaan kuten $|000...\rangle$
3. Pitkät koherenssiajat, paljon pidemmät kuin porttioperaatioiden kesto
4. Universaali kvanttiporttijoukko
5. Tarkka kubitin tilan mittaaminen

Kriteerit ohjaavat universaalin kvanttitietokoneen toteututtamiseen tähtäävää tutkimusta. Suprajohteisiin, optiikkaan ja puolijohteisiin perustuvat kokeelliset toteutukset ovat osoittautuneet lupaaviksi.

Suprajohteisiin perustuvissa kvanttitietokoneissa kubitit on toteutettu Joshepsonin liitoskilla kytkettyinä virtapiireinä[23]. Kubitin tiloja $|0\rangle$ ja $|1\rangle$ kuvaavat Cooperin parien lukumäärä, indusoitu magneettivuo tai virran oskillaation energiatilat [38]. Kubitteja ohjataan mikroaalloilla ja magneettikentällä. Tietokoneessa kubitit asetellaan ristikoksi, jossa kubitille voidaan toteuttaa kahden kubitin kvanttiportteja viereisten kubittien kanssa. Suprajohtavat kubitit ovat herkkiä dekoherenssille ja toimiakseen ne on jäähdytettävä lähelle aboluuttista nollapistettä. [23]

Toinen lupaava toteutustapa perustuu optiikkaan. Optisissa kvanttitietokoneissa kubitteina käytetään fotoneita. Kubitin kahta tilaa vastaavat esimerkiksi fotonin polarisaatio [39]

tai fotonien lukumäärä [13]. Fotoni ei ole herkkä dekoherenssille ja porttioperaatiot voidaan toteuttaa yleisesti käytetyillä lineaarisen optiikan elementeillä. Optisen kvanttitietokoneen ohjelmoitavuus on kuitenkin haaste, sillä jokainen laskentapiiri on toteuttava fyysisesti optisista elementeistä. [13]

Puolijohteisiin perustuvissa kvanttitietokoneissa kubitit on toteutettu kvanttipisteisiin vangittuina elektroneina. Esimerkiksi elektronin spinin avulla voidaan magneettikentässä muodostaa kahden energiatilan systeemi, jonka ohjaus onnistuu Rabi-oskillaatiolla. Puolijohdekomponentteihin pohjautuva kvanttipistekubitit ovat lähellä nykyistä puolijohdeteknologiaa, joten ne ovat valmistettavissa vastaavilla prosesseilla kuin nykyiset tietokoneet, mikä on lisännyt kvanttipistekubitteihin kohdistuvaa kiinnostusta. [36]

6.2 Kvanttitietokoneen rakentamisen haasteet

Kvanttitietokoneiden kehityksen kannata suurin haaste on dekoherenssin aiheuttama virhe. Kubitit ovat usein atomaarisia systeemeitä, jolloin heikkokin vuorovaikutus ympäristön kanssa altistaa laskennan virheille. Virheiden määrä kasautuu käytettävien kubittien ja operaatioiden määrän kasvaessa. Virheenkorjausalgoritmien avulla voidaan laskea kubittien tarkkuuden vaatimusta, mutta vain rajallisesti. [22]

On esitetty, että topologinen kvanttilaskenta voisi olla ratkaisu virheenkorjaukseen. Topologisista kubiteista voidaan tehdä kooltaan suuria, jolloin kvantti-informaatio ei ole yhtä altis lokaaleille virheille. [40] Topologisilla kvanttitietokoneella ei ole toistaiseksi fyysistä toteutusta, mutta tutkimustyötä sitä kohti tehdään [22].

6.3 Kehityksen saavutukset

Tällä hetkellä kehittyneimmät kvanttitietokoneet edustavat niin kutsuttua NISQ-teknologiaa (*Noisy Intermediate-Scale Quantum*). NISQ-kvanttitietokoneet ovat kooltaan keskikokoisia, muutamista kymmenistä satoihin kubitteihin. Niillä suoritetuissa operaatioissa esiintyy virhettä, joka kumuloituu piirien kompleksisuuden kasvaessa, minkä takia ne eivät sovellu universaaliin kvanttilaskentaan.[22]

Suurimpana kvanttilaskennan saavutuksena voidaan pitää vuonna 2020 ensimmäisen kerran saavutettua kvanttiherruutta. Kiinalainen tutkimusryhmä ratkaisi 76:n kubitin optisella kvanttitietokoneella 200 sekunissa ongelman, johon maailman tehokkaimmalla supertietokoneella olisi kulunut 0,6 miljoonaa vuotta [41]. Vastaavissa suorituskyvyn osoituksissa tehtävät ovat toistaiseksi olleet juuri käytettävän kvanttitietokoneen fyysiselle toteutukselle räätälöityjä ongelmia, joiden ratkaisulla ei ole käytännön hyötyä. Klassisilla tietokoneilla on pyritty simuloimaan kvanttitietokoneen toimintaa. Myös Googlen tutkijat raportoivat saavuttaneensa kvanttiherruuden vuonna 2019 [23], mutta väite on myöhemmin kiistetty [42].

6.4 Kvanttilaskennan mahdolliset sovellukset

Kvanttilaskennalle on löydetty useita käytönnönhyötyä tuovia sovelluskohteita. Kvanttitietokone toisi suuren edun esimerkiksi simulaatio- ja optimointiongelmissa sekä kryptografiassa ja koneoppimisessa. [22]

Universaalin kvanttitietokoneen avulla monimutkaisten molekyylien ja reaktioiden mallintaminen tarkasti olisi mahdollista, mikä olisi esimerkiksi laskennallisen fysiikan, kemian ja biologian tutkimuksen kannalta mullistavaa. Myös HHL-algoritmi ja amplitudin vahvistamisen avulla on mahdollisuus suorittaa hyvinkin kompleksisten systeemien simulaatioita ja optimontia.[9] Optimointiongelmissa voidaan saavuttaa myös adiabaattisella kvanttijäähdytyksellä suuri etu klassisiin menetelmiin nähden. Sille on optimointiongemissa esitetty käyttökohteita laajasti taloustieteistä biokemiaan.[22]

Shorin esitettyä faktorointialgoritmin on ollut selvää että universaalilla kvanttitietokoneella olisi kryptografisia sovelluksia [25]. Shorin ja Groverin algoritmeilla olisi nykypäivän kryptografiset suojaukset rikottavissa. Kvantti-informaatiotekniikka mahdollistaisi kuitenkin aiempaa vahvempia suojausmenetelmiä [25]. Esimerkiksi lomittuneita fotoneita, "lentäviä kubitteja", hyödyntämällä voitaisiin teoriassa toteuttaa viestinnän suojaus, jota voidaan pitää täysin murtamattomana. [22]

Kvanttilaskennalla voisi saavuttaa suurta hyötyä koneoppimisessa, jossa statistisin menetelmin käsitellään hyvin suuria datamääriä. Useille koneoppimiseen käytetyille algoritmeille on kehitetty vastaavia kvanttialgoritmeja, joilla saavutetaan huomattava nopeudenlisäys. Pienelläkin virheettömällä kvanttitietokoneella voitaisiin ajaa koneoppimisalgoritmeja suuressa laskenta-avaruudessa tuottaen arvokasta informaatiota. Esimerkiksi ihmisen genomi olisi talletettavissa noin 34:een kubittiin. Koneoppimisen kannalta erityisen kriittiseksi pullonkaulaksi nousee kuitenkin rajapinta kvantti-informaation ja klassisen informaation välillä. [9]

Edellä mainitut sovellukset vaatisivat toimiakseen lähes täysin virhesietoisen kvanttitietokoneen, minkä takia ne ovat toistaiseksi saavuttamattomissa. Ensimmäisen hyödyllisen kvanttilaskennan käyttökohteen odotetaan olevan kvanttimekaanisen systeemin analoginen simulaatio. Se ei vaadi läheskään yhtä suurta laskennallista syvyyttä, kuin esimerkiksi HHL-algoritmi tai Shorin algoritmi, mutta on klassisiin menetelmiin nähden huomattavasti tehokkaampaa. Kvanttihyödyn tuova kvanttisimulaatio olisi toteutettavissa NISQ-laitteistolla, joka operoi noin 100:n kubitin suuruusluokassa. [22] On esitetty, että kvanttitietokoneiden kehitys voisi tulevaisuudessa seurata kvanttiversiota Mooren laista, jolloin laskentakapasiteeti kaksinkertaistuisi muutaman vuoden välein [23].

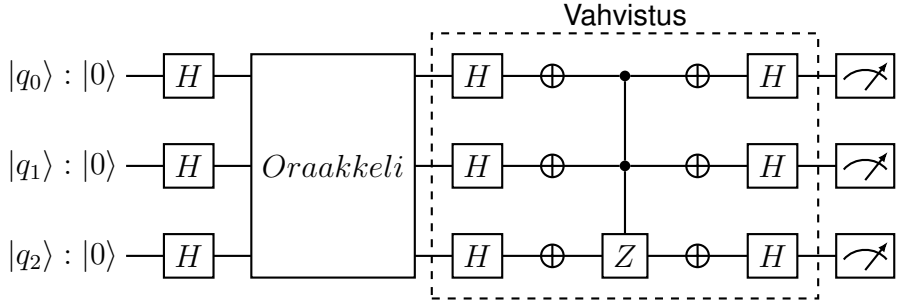
Kvanttilaskenta on mittaamista ja virheitä lukuun ottamatta reversiibeli prosessi [11]. Kvanttilaskenta ei tuota laskennan ohessa hukkalämpöä kuten supertietokoneet, minkä surauksena, energiatehokkuudessa voidaan kvanttilaskennalla saavuttaa suuria hyötyjä klassiseen laskentaan nähden.

7. GROVERIN ALGORITMIN KOKEELLINEN TOTEUTUS

Työssä Groverin algoritmi toteutettiin kolmella kubitilla, $n = 3$. Algoritmi vastaa hakua tietokannasta, jonka koko N on $2^n = 8$. Groverin algoritmilla haetaan yhtä tulosta yhdellä iteraatiolla. Algoritmi ajetaan simulaatiolla sekä kvanttitietokoneella *IBM Q Experience* -alustalla.

7.1 Toteutus

Kolmen kubitin Groverin algoritmin yksi mahdollinen toteutus on esitetty kuvassa 7.1. Superpositiotilan alusteminen suoritetaan kohdistamalla Hadamardin portti jokaiselle kubitille. Työssä käytetään tilan $|111\rangle$ todennäköisyysamplitudin kääntävää oraakkelia. $|111\rangle$-oraakkelipiiri voidaan esittää kahdella kubitilla kontrolloituna Pauli-Z-porttina, CCZ-porttina. [28]



Kuva 7.1. *Kolmen kubitin Groverin algoritmi yhdellä iteraatiolla, perustuen [28]*

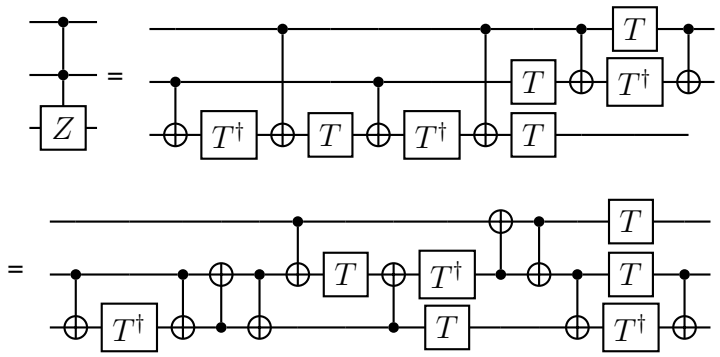
Laskennallisesti yhden iteraation jälkeen etsityn arvon mittaamisen todennäköisyys on $\frac{5}{4\sqrt{2}}$ = 78,125 %. Vastaava klassinen algorimti koostuisi yhdestä etsinnästä tietokannasta, sekä sitä seuraavasta satunnaisesta arvauksesta, jos haettu arvo ei löytynyt. Tälläisellä algoritmilla voidaan parhaimmillaan saavuttaa $\frac{1}{8} + \frac{7}{8} \cdot \frac{1}{7}$ = 25 %:n todennäköisyys haetun arvon löytymiselle.[28]

Algoritmi ajettiin *IBM Quantum Experience* -alustalla [43], joka on avoin kvanttilaskentaan tarkoitettu pilvipalvelu. Alustalla on mahdollisuus ohjelmoida kvanttipiirejä, ja ajaa niitä etäyhteydellä IBM:n simulaattoreilla sekä muutaman kubitin kvanttitietokoneilla. Kvanttialgoritmien ohjelmointiin on kehitetty useita ohjelmointikieliä, joista palvelussa on käytössä OpenQASM ja Qiskit. Lisäksi alustalla yksinkertaisia kvanttipiirejä voi toteuttaa graafi-

sella *Quantum Composer* -työkalulla. Tässä työssä käytettiin 5:n kubitin *ibmq_santiago*-kvanttitietokonetta, joka on yksi *IBM Quantum Falcon r4* -prosessoreista. IBM:n kvanttitietokoneet perustuvat suprajohteisiin. Piirin klassinen simulaatio suoritettiin samalla alustalla käytettäen *ibmq_qasm*-simulaattoria.

IBM:n suprajohtaville kubiteille käytössä yhden kubitin porteista $\sqrt{\text{X}}$, X sekä kaikki vaiheensiirtoportit. Algoritmin vaatima Hadamardin portti voidaan rakentaa yhden $\sqrt{\text{X}}$-portin ja kahden S-portin avulla. Kahdelle kubitille käytettävissä on ainoastaan CNOT-portti. CCZ-portti voidaan konstruoida käyttäen vain yhden kubitin vaiheensiirtoportteja sekä kahden kubitin CNOT-portteja [44]. Toteutus on esitetty kuvassa 7.2. Käytössä olleessa arkkitehtuurissa kubitit ovat rinnakkain siten, että kubiteille voi kohdistaa CNOT-portteja ainoastaan vierekkäisten kubittien kanssa. Tämän vuoksi kubiteille q_0 ja q_2 ei voida suorittaa CNOT-porttia, minkä vuoksi CCZ-portin rakentamiseksi on käytetyllä laitteistolla suoritettava suurempi määrää CNOT-portteja. Myös tämä toteutus on esitetty kuvassa 7.2.

Lopullinen kuvan 7.1 piirin kvanttitietokoneelle ohjelmoitu toteutus sisälsi yhteensä 9 $\sqrt{X}$-porttia, 35 vaiheensiirtoporttia ja 20 CNOT-porttia. Toteutus on esitetty kokonaisuudessaan OpenQASM-ohjelmointikielellä liitteessä A. Suurin virhe käytetyllä kvanttitietokonella syntyy CNOT-portin ja mittauksen epätarkkuudesta. Käytetylle laitteistolle IBM ilmoittaa yhden CNOT-porttioperaation virheen olevan 0,60 % ja mittauksen 1,4 %.

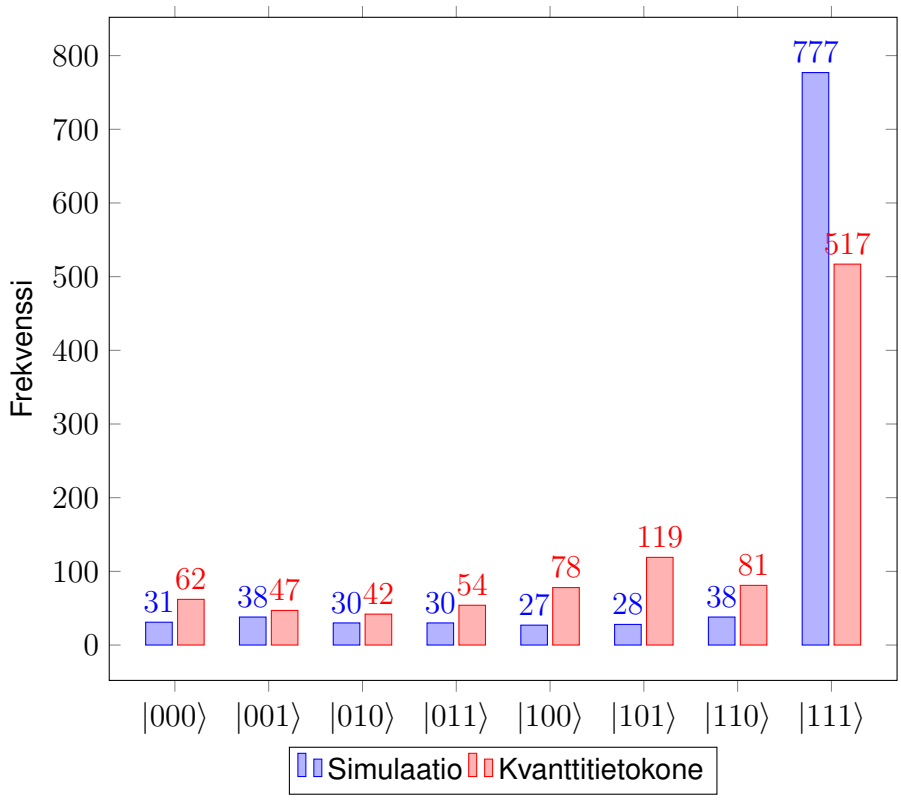


Kuva 7.2. *Piiri kolmen kubitin CCZ-portille käyttäen yhden kubitin vaiheportteja sekä kahden kubitin CNOT-portteja*

7.2 Tulokset

Algoritmi ajettiin 1000 kertaa sekä simulaationa että kvanttitietokoneella. Saadut tulokset on esitetty kuvassa 7.3.

Simulaatiolla löydettiin haettu tulos 777 kertaa 1000:sta, mikä vastaa hyvin laskennallista todennäköisyyttä. Tulosta voidaan pitää osoituksena piirin onnistuneesta toteutuksesta. Kvanttitietokoneella ajetun algoritmilla löydettiin haettu arvo 517 kertaa 1000:sta, mikä vastaa 51,7 %:n todennäköisyyttä. On myös huomionarvoista, että haetun arvon fre-



Kuva 7.3. *Kolmen kubitin Groverin algoritmin 1000:n ajon tulokset klassisella simulaatiolla ja kvanttitietokoneella*

kvenssi oli mittaustuloksissa yli 4 kertaa suurempi kuin seuraavaksi yleisimmän mittaustuloksen. Todennäköisyys on simulaatiota huomattavasti pienempi dekoherenssin aiheuttaman virheen seurauksena. Virheestä huolimatta kvanttitietokoneella saavutettiin suurempi todennäköisyys kuin vastaavalla klassisella algoritmilla.

8. YHTEENVETO

Työssä tutkittiin kirjallisuudesta, kuinka kvanttilaskennallinen prosessi rakentuu yksinkertaisista kvanttimekaanisista operaatioista tehokkaiksi kvanttialgoritmeiksi, sekä mitä kvanttilaskennalla voidaan saavuttaa. Lisäksi kvanttilaskennan tehokkuus osoitettiin käytännön demonstraatiolla.

Kvanttilaskennalla olisivat monet supertietokoneille mahdottomat ongelmat ovat ratkaistavissa. Löydetyillä kvanttialgoritmeilla voidaan simuloida kvanttimekaanisia järjestelmiä, ratkaista kompleksisia optimointiongelmia sekä ajaa koneoppimisalgoritmeja parhaimmillaan eksponentiaalisesti klassisia supertietokoneita nopeammin. Skaalautuvan ja virheensietoisen kvanttitietokoneen vaikutus olisi mullistava monella tieteenalalla. Suuren laskentatehokkuuden lisäksi kvanttilaskenta mahdollistaa ongelmien ratkaisemisen hyvin energiatehokkaasti.

Uusien kvanttialgoritmien kehittäminen ei ole yksinkertaista kvantti-informaation epädeterministisen luonteen vuoksi. On todennäköistä, että löydetyt kvanttialgoritmit muodostavat vain pienen osan mahdollisista käyttökohteista, ja että kvanttilaskennan todellinen potentiaali tarkentuu tulevaisuudessa.

Kvanttilaskennan teorian tutkimus on edennyt pidemmälle kuin kvanttitietokoneiden fyysinen toteutus. Ensimmäisiä kvanttitietokoneita on rakennettu ja niiden tehokkuus on osoitettu kokeellisesti. Ensimmäiset kvanttitietokoneiden hyödylliset käyttökohteet eivät ole kaukana, mutta universaali kvanttilaskenta on nykyisellä tekniikalla vielä toistaiseksi saavuttamattomissa. Tehokkaiden kvanttitietokoneiden rakentaminen on hyvin haastavaa, sillä kubitit ovat herkkiä ympäristön vuorovaikutuksen aiheuttamille virheille.

Viime vuosina on avautunut ensimmäistä kertaa mahdollisuus käyttää todellisia kvanttitietokoneita etäyhteydellä. Avoimet kvanttitietokoneet ovat laskentakapasiteetiltaan pieniä ja niissä esiintyy paljon virhettä. Niillä on kuitenkin mahdollista toteuttaa yksinkertaisilla kvanttialgoritmeilla laskutoimituksia, jotka osoittavat kvanttilaskennassa piilevän tehokkuuden. Työssä tämä tehtiin toteuttamalla Groverin algoritmi yhdellä iteraatiolla. Kvanttitietokoneella löydettiin haettu arvo yhdellä iteraatiolla 51,7 %:n todennäköisyydellä. Klassisella tietokoneella vastaavan algoritmin todennäköisyys voisi yltää parhaimmillaan 25 %:iin.

LÄHTEET

[1] Neumann, J. von, Wheeler, N. A. ja Beyer, R. T. *Mathematical foundations of quantum mechanics, new edition*. Princeton: Princeton University Press, 2018. ISBN: 1400889928.

[2] Bennett, C. ja Shor, P. Quantum information theory. eng. *IEEE transactions on information theory* 44.6 (1998), s. 2724–2742. ISSN: 0018-9448.

[3] Le Bellac, M. *A short introduction to quantum information and quantum computation*. eng. Cambridge, UK ; Cambridge University Press, 2006. ISBN: 1-107-16766-3.

[4] Shor, P. Algorithms for quantum computation: discrete logarithms and factoring. eng. *Proceedings 35th Annual Symposium on Foundations of Computer Science*. IEEE Comput. Soc. Press, 1994, s. 124–134. ISBN: 0818665807.

[5] Einstein, A., Podolsky, B. ja Rosen, N. Can Quantum-Mechanical Description of Physical Reality Be Considered Complete? eng. *Physical review* 47.10 (1935), s. 777–780. ISSN: 0031-899X.

[6] Bell, J. S. On the Einstein Podolsky Rosen paradox. eng. *Physics (New York. 1964)* 1.3 (1964), s. 195–200. ISSN: 0554-128X.

[7] Yin, J., Cao, Y., Li, Y.-H., Liao, S.-K., Zhang, L., Ren, J.-G., Cai, W.-Q., Liu, W.-Y., Li, B., Dai, H., Li, G.-B., Lu, Q.-M., Gong, Y.-H., Xu, Y., Li, S.-L., Li, F.-Z., Yin, Y.-Y., Jiang, Z.-Q., Li, M., Jia, J.-J., Ren, G., He, D., Zhou, Y.-L., Zhang, X.-X., Wang, N., Chang, X., Zhu, Z.-C., Liu, N.-L., Chen, Y.-A., Lu, C.-Y., Shu, R., Peng, C.-Z., Wang, J.-Y. ja Pan, J.-W. Satellite-based entanglement distribution over 1200 kilometers. eng. *Science (American Association for the Advancement of Science)* 356.6343 (2017), s. 1140–1144. ISSN: 0036-8075.

[8] Jerome, J. W. ja Polizzi, E. Discretization of time-dependent quantum systems: real-time propagation of the evolution operator. eng. *Applicable analysis* 93.12 (2014), s. 2574–2597. ISSN: 0003-6811.

[9] Outeiral, C., Strahm, M., Shi, J., Morris, G. M., Benjamin, S. C. ja Deane, C. M. The prospects of quantum computing in computational molecular biology. eng. *Wiley interdisciplinary reviews. Computational molecular science* 11.1 (2021). ISSN: 1759-0876.

[10] Turing, A. M. On Computable Numbers, with an Application to the Entscheidungsproblem. eng. *Proceedings of the London Mathematical Society* s2-42.1 (1937), s. 230–265. ISSN: 0024-6115.

[11] Deutsch, D. Quantum Theory, the Church-Turing Principle and the Universal Quantum Computer. eng. *Proceedings of the Royal Society of London. Series A, Mathematical and physical sciences* 400.1818 (1985), s. 97–117. ISSN: 1364-5021.

[12] Aharonov, D., Van Dam, W., Kempe, J., Landau, Z., Lloyd, S. ja Regev, O. Adiabatic quantum computation is equivalent to standard quantum computation. eng. *SIAM journal on computing* 37.1 (2007). ISSN: 0097-5397.

[13] Krovi, H. Models of optical quantum computing. eng. *Nanophotonics (Berlin, Germany)* 6.3 (2017), s. 531–541. ISSN: 2192-8606.

[14] Nayak, C., Simon, S. H., Stern, A., Freedman, M. ja Das Sarma, S. Non-Abelian anyons and topological quantum computation. eng. *Reviews of modern physics* 80.3 (2008), s. 1083–1159. ISSN: 0034-6861.

[15] Raussendorf, R., Browne, D. E. ja Briegel, H. J. Measurement-based quantum computation on cluster states. eng. *Physical review. A, Atomic, molecular, and optical physics* 68.2 (2003). ISSN: 1050-2947.

[16] Raussendorf, R., Briegel, H. J., Browne, D. E., Dür, W. ja Van den Nest, M. Measurement-based quantum computation. eng. *Nature physics* 5.1 (2009), s. 19–26. ISSN: 1745-2473.

[17] Chi-Chih Yao, A. Quantum circuit complexity. eng. *Proceedings of 1993 IEEE 34th Annual Foundations of Computer Science*. IEEE, 1993, s. 352–361. ISBN: 9780818643705.

[18] Molina, A. ja Watrous, J. Revisiting the simulation of quantum Turing machines by quantum circuits. eng. *Proceedings of the Royal Society. A, Mathematical, physical, and engineering sciences* 475.2226 (2019), s. 20180767–20180767. ISSN: 1364-5021.

[19] Brown, J. *Kvanttitietokone*. fin. Helsinki: Terra cognita, 2000. ISBN: 952-5202-42-9.

[20] Törmä, P. ja Suominen, K.-A. Kvanttitietokoneet: Teoria ja käytäntö. fin. *Arkhimedes* 4 (1995).

[21] Cleve, R., Ekert, A., Macchiavello, C. ja Mosca, M. Quantum algorithms revisited. eng. *Proceedings of the Royal Society. A, Mathematical, physical, and engineering sciences* 454.1969 (1998), s. 339–354. ISSN: 1364-5021.

[22] Preskill, J. Quantum Computing in the NISQ era and beyond. eng. *Quantum* 2 (2018), s. 79–.

[23] Quantum supremacy using a programmable superconducting processor. eng. *Nature (London)* 574.7779 (2019), s. 505–510. ISSN: 0028-0836.

[24] Harrow, A. W., Hassidim, A. ja Lloyd, S. Quantum algorithm for linear systems of equations. eng. *Physical review letters* 103.15 (2009), s. 150502–150502. ISSN: 0031-9007.

[25] Törmä, P. Kvanttitietokoneet ja kvanttikryptografia. fin. *Arkhimedes* 5 (2000).

[26] Coppersmith, D. An approximate Fourier transform useful in quantum factoring. eng (2002).

[27] Grover, L. K. A fast quantum mechanical algorithm for database search. eng (1996).

[28] Figgatt, C., Maslov, D., Landsman, K. A., Linke, N. M., Debnath, S. ja Monroe, C. Complete 3-Qubit Grover search on a programmable quantum computer. eng. *Nature communications* 8.1 (2017), s. 1918–9. ISSN: 2041-1723.

[29] Viamontes, G., Markov, I. ja Hayes, J. Is quantum search practical? eng. *Computing in science & engineering* 7.3 (2005), s. 62–70. ISSN: 1521-9615.

[30] Grover, L. K. A framework for fast quantum mechanical algorithms. eng (1997).

[31] Quantum Amplitude Amplification and Estimation. eng (2000).

[32] Wiebe, N., Braun, D. ja Lloyd, S. Quantum algorithm for data fitting. eng. *Physical review letters* 109.5 (2012), s. 050505–050505. ISSN: 0031-9007.

[33] Lloyd, S. Universal Quantum Simulators. eng. *Science (American Association for the Advancement of Science)* 273.5278 (1996), s. 1073–1078. ISSN: 0036-8075.

[34] Laflamme, R., Laflamme, R., Miquel, C., Miquel, C., Paz, J. P., Paz, J. P., Zurek, W. H. ja Zurek, W. H. Perfect quantum error correcting code. eng. *Physical review letters* 77.1 (1996), s. 198–201. ISSN: 0031-9007.

[35] Shor, P. W. Scheme for reducing decoherence in quantum computer memory. eng. *Physical review. A, Atomic, molecular, and optical physics* 52.4 (1995), R2493–R2496. ISSN: 1050-2947.

[36] Veldhorst, M., Yang, C. H., Hwang, J. C. C., Huang, W., Dehollain, J. P., Muhonen, J. T., Simmons, S., Laucht, A., Hudson, F. E., Itoh, K. M., Morello, A. ja Dzurak, A. S. A two-qubit logic gate in silicon. eng. *Nature (London)* 526.7573 (2015), s. 410–414. ISSN: 0028-0836.

[37] DiVincenzo, D. P. The Physical Implementation of Quantum Computation. eng. *Fortschritte der Physik* 48.9-11 (2000), s. 771–783. ISSN: 0015-8208.

[38] You, J. ja Nori, F. Superconducting circuits and quantum information. eng. *Physics today* 58.11 (2005), s. 42–47. ISSN: 0031-9228.

[39] Martín-López, E., Laing, A., Lawson, T., Alvarez, R., Zhou, X.-Q. ja O'Brien, J. L. Experimental realization of Shor's quantum factoring algorithm using qubit recycling. eng. *Nature photonics* 6.11 (2012), s. 773–776. ISSN: 1749-4885.

[40] Kitaev, A. Anyons in an exactly solved model and beyond. eng. *Annals of physics* 321.1 (2006), s. 2–111. ISSN: 0003-4916.

[41] Zhong, H.-S., Wang, H., Deng, Y.-H., Chen, M.-C., Peng, L.-C., Luo, Y.-H., Qin, J., Wu, D., Ding, X., Hu, Y., Hu, P., Yang, X.-Y., Zhang, W.-J., Li, H., Li, Y., Jiang, X., Gan, L., Yang, G., You, L., Wang, Z., Li, L., Liu, N.-L., Lu, C.-Y. ja Pan, J.-W. Quantum computational advantage using photons. eng. *Science (American Association for the Advancement of Science)* 370.6523 (2020), s. 1460–1463. ISSN: 0036-8075.

[42] Pednault, E., Gunnels, J. A., Nannicini, G., Horesh, L. ja Wisnieff, R. Leveraging Secondary Storage to Simulate Deep 54-qubit Sycamore Circuits. eng (2019).

[43] *IBM Quantum*. URL: `https://quantum-computing.ibm.com/`. viitattu 14.5.2021.

[44] Barnes, E., Arenz, C., Pitchford, A. ja Economou, S. E. Fast microwave-driven three-qubit gates for cavity-coupled superconducting qubits. eng. *Physical review. B* 96.2 (2017). ISSN: 2469-9950.

LIITE A: SUORITETTU KVANTTIALGORITMI

```
OPENQASM 2.0;
include "qelib1.inc";

qreg q[5];

creg c[3];


rz(1.5707963267948961) q[0];
sx q[0];
rz(1.5707963267948966) q[0];
rz(1.5707963267948961) q[1];
sx q[1];
rz(1.5707963267948966) q[1];
rz(1.5707963267948961) q[2];
sx q[2];
rz(1.5707963267948966) q[2];
barrier q[0], q[1], q[2];
cx q[1], q[2];
rz(5.497787143782137) q[2];
cx q[1], q[2];
cx q[2], q[1];
cx q[1], q[2];
cx q[0], q[1];
rz(0.7853981633974487) q[1];
cx q[2], q[1];
rz(5.497787143782137) q[1];
cx q[1], q[0];
cx q[0], q[1];
rz(0.7853981633974487) q[0];
rz(0.7853981633974487) q[2];
cx q[1], q[2];
rz(0.7853981633974487) q[1];
rz(5.497787143782137) q[2];
cx q[1], q[2];
barrier q[0], q[1], q[2];
rz(4.71238898038469) q[0];
sx q[0];
rz(1.5707963267948966) q[0];
rz(4.71238898038469) q[1];
sx q[1];
rz(1.5707963267948966) q[1];
rz(4.71238898038469) q[2];
sx q[2];
rz(1.5707963267948966) q[2];
cx q[1], q[2];
rz(5.497787143782137) q[2];
cx q[1], q[2];
cx q[2], q[1];
cx q[1], q[2];
cx q[0], q[1];
rz(0.7853981633974487) q[1];
cx q[2], q[1];
rz(5.497787143782137) q[1];
cx q[1], q[0];
cx q[0], q[1];
rz(0.7853981633974487) q[0];
rz(0.7853981633974487) q[2];
cx q[1], q[2];
rz(0.7853981633974487) q[1];
rz(5.497787143782137) q[2];
cx q[1], q[2];
barrier q[0], q[1], q[2];
rz(1.5707963267948966) q[0];
sx q[0];
rz(4.71238898038469) q[0];
rz(1.5707963267948966) q[1];
sx q[1];
rz(4.71238898038469) q[1];
rz(1.5707963267948966) q[2];
sx q[2];
rz(4.71238898038469) q[2];
measure q[0] -> c[0];
measure q[1] -> c[1];
measure q[2] -> c[2];
```