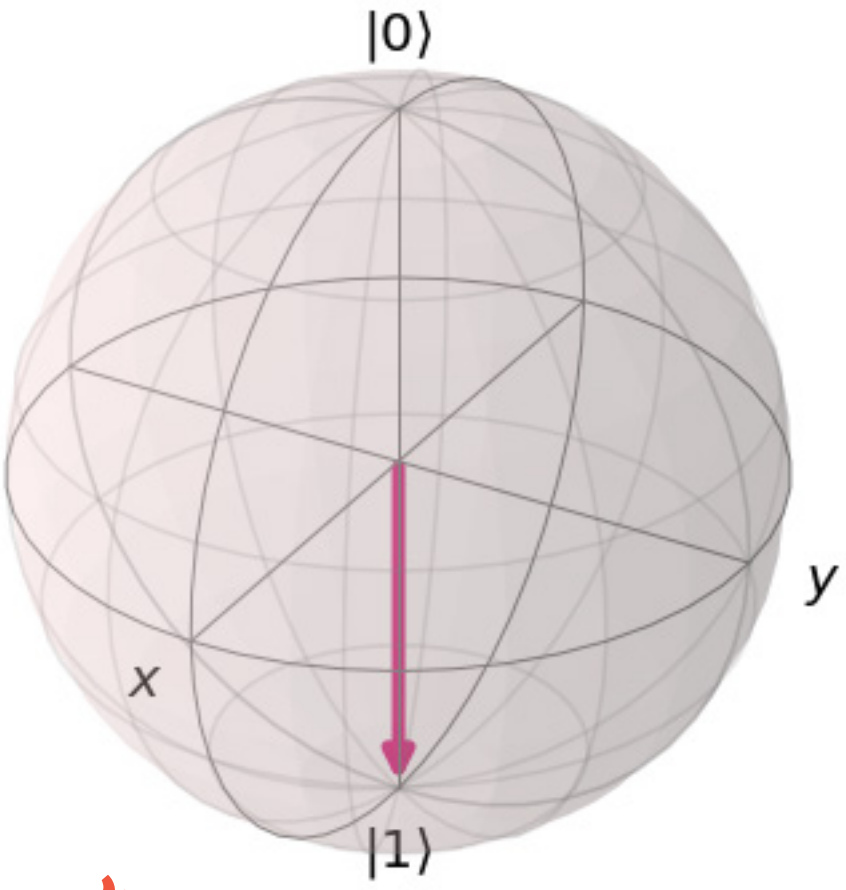
Informaatioteknologian tiedekunnan julkaisuja
No. 88/2021

Henri Heinonen

Katsaus kvanttilaskentateknologiaan ja sen sovelluksiin



JYVÄSKYLÄN YLIOPISTO
INFORMAATIOTEKNOLOGIAN
TIEDEKUNTA

Informaatioteknologian tiedekunnan julkaisuja
No. 88/2021

Editor: Pekka Neittaanmäki
Covers: Kati Valpe

Copyright © 2021
Henri Heinonen ja Jyväskylän yliopisto

ISBN 978-951-39-8534-9 (verkkoj.)
ISSN 2323-5004

Jyväskylä 2021

Esipuhe

Tämä raportti on katsaus kvanttitietokoneiden nykytilaan ja lähitulevaisuuteen erityisesti ohjelmiston näkökulmasta. Se on saanut paljon inspiraatiota kvanttitietokoneita kriittisesti ja skeptisesti käsittelevästä raportista (National Academies of Sciences, Medicine ym. 2019) ja kirjasta (Dyakonov 2020), mutta mukana on myös tietoa, joka on peräisin kvanttitietokoneiden ohjelmointia käsittelevistä kirjoista, artikkeleista ja videoista.

Kvanttilaskenta ja meille tutumpi klassinen laskenta eivät ole kaksi eri tiedonalaa. Laskennan perusmuoto on kvanttilaskenta ja laskennan perusyksikkö onkin bitin sijasta kvanttibitti eli kubitti. (Bernhardt 2019) Kvanttibitti voi olla joko 0 tai 1, mutta se voi olla myös kombinaatio näistä molemmista (superpositio). Klassisen tietokoneen tilan määrää bittien binääriarvo. Vastaavan määrän kubitteja sisältävä kvanttitietokone voi olla kaikissa mahdollisissa vastaavan klassisen tietokoneen tiloissa mahdollistaen paljon suuremman ongelma-avaruuden. Tämä avaruuden hyödyntäminen tosin vaatii kubittien sisäisen yhteyden (lomittumisen), hyvän eristyksen ulkopuolisesta ympäristöstä ja hyvin tarkan kontrollin. (National Academies of Sciences, Medicine ym. 2019)

Raportin liitteet A: “Kvanttilaskennan trendit vuodelle 2020 ja eteenpäin”, B: “Kvanttialgoritmit ja kvanttisovellukset”, C: “Kvanttilaskennan kurssit, tutkinnot ja tutkimusryhmät” ja D: “Kvanttilaskenta ja digitaalinen turvallisuus” ovat lyhyitä yhteenvetoja materiaaleista, jonka ovat laatineet yhteistyössä Anthony Ogbechie ja professori Pekka Neittaanmäki Jyväskylän yliopistolta. Kiitos Pekalle ja Anthonylle arvokkaiden materiaalien koostamisesta.

Raportin muut osat ovat Jyväskylän yliopiston projektitutkijan Henri Heinosen työstämiä.

Antti-Juhani Kaijanaholle suurkiitos pro graduille tarkoitetusta LaTeX-mallipohjasta (gradu3.cls), jota muokkasin tähän raporttiin soveltuvaan muotoon.

Jyväskylä, 5. helmikuuta 2021

Henri Tapani Heinonen

Termiluettelo

Aaltohiukkasdualismi	Kvanttioliolla on tyypillisesti sekä aalto- että hiukkastyyppisiä ominaisuuksia. Vaikka järjestelmän kehitys seuraa aaltoyhtälöä, mikä tahansa mittaus palauttaa arvon, joka on johdonmukainen sen kanssa, että kyseessä on hiukkanen. (National Academies of Sciences, Medicine ym. 2019)
Bitti	Bitti (engl. 'bit') eli binäärinen numero (engl. 'binary digit') on kaksiarvoista logiikkaa käyttävän klassisen tietokoneen tiedon perusosanen.
Koherenssi	Kun kvanttijärjestelmän tilan voi kuvata kompleksilukujen joukkona, yksi kullekin järjestelmän perustilalle, järjestelmän tilan sanotaan olevan "koherentti". Koherenssi on välttämätön kvantti-ilmiöille kuten kvantti-interferenssille, superpositiolle ja lomittumiselle. Pienet vuorovaikutukset ympäristön kanssa saavat kvanttijärjestelmät hiljalleen dekoheroituvaan. Ympäristön vuorovaikutukset tekevät jopa kompleksiset kertoimet probabilisistisiksi jokaiselle tilalle. (National Academies of Sciences, Medicine ym. 2019)
Kubitti	Kvanttibitti (engl. 'quantum bit') eli kubitti (engl. 'qubit') on kaksiarvoista logiikkaa käyttävän kvanttitietokoneen tiedon perusosanen.
Kutritti	Kvanttitritti (engl. 'quantum trit') eli kutritti (engl. 'qutrit') on kolmiarvoista logiikkaa käyttävän kvanttitietokoneen tiedon perusosanen.
Lomittuminen	Lomittuminen eli kietoutuminen on joidenkin (mutta ei kaikkien) monihiukkassuperpositiotilojen erikoisominaisuus, jossa yhden hiukkasen tilan mittaus romahduttaa muiden hiukkasten tilan, jopa kun hiukkaset ovat kaukana toisistaan ilman mitään havaittavaa keinoa vuorovaikutukseen. Tämä syntyy, kun eri hiukkasten aaltofunktiot eivät ole separoituvissa (matemaatti-

	sin termein, kun koko järjestelmän aaltofunktiota ei voi kirjoittaa jokaisen hiukkasen aaltofunktioiden tulona). Tälle ilmiölle ei ole klassista analogiaa. (National Academies of Sciences, Medicine ym. 2019)
Mittaus	Kvanttijärjestelmän mittaus fundamentaalisesti muuttaa sitä. Tapauksessa, jossa mittaus tuottaa hyvin määritellyn arvon, järjestelmä jää tilaan, joka vastaa mitattua arvoa. Tätä yleisesti kutsutaan "aaltofunktion romahduttamiseksi". (National Academies of Sciences, Medicine ym. 2019)
Superpositio	Kvanttijärjestelmä voi olla kahdessa tai useammassa tilassa samanaikaisesti, mihin viitataan tilojen "superpositiona"tai "superpositiotilana". Sellaisen superpositiotilan aaltofunktiota voidaan kuvailla kontribuoivien tilojen lineaarisena kombinaationa, jossa on kompleksiset kertoimet. Nämä kertoimet kuvaavat magnitudia ja kontribuoivien tilojen välistä suhteellista vaihetta. (National Academies of Sciences, Medicine ym. 2019)
Tritti	Tritti (engl. 'trit') eli ternäärinen numero (engl 'ternary digit') on kolmiarvoista logiikkaa käyttävän klassisen tietokoneen tiedon perusosanen.

Kuviot

Kuvio 1. Kubittitila $|0\rangle$ kuvattuna Blochin pallolla 4
Kuvio 2. Kubittitila $|1\rangle$ kuvattuna Blochin pallolla 4
Kuvio 3. Bellin tilan zz-mittaukset 6
Kuvio 4. Bellin tila ja zz-mittaukset kubiteille 11
Kuvio 5. Bellin tilan zz-mittausten histogrammi 11
Kuvio 6. Bernstein–Vazirani-algoritmin toteuttava kvanttipiiri 14
Kuvio 7. Kvanttipiiri, joka tuottaa satunnaisen bitin 28
Kuvio 8. IBM:n kvanttitietokoneen tuottama satunnaisbittijakauma 28
Kuvio 9. Majoriteettipiiri 48

Taulukot

Taulukko 1. Klassisen algoritmin askeleet salaisen luvun arvaamiseksi. 13
Taulukko 2. 320 kappaletta satunnaislukuja kvanttisimulaattorista 33
Taulukko 3. 320 kappaletta RIPEMD-160-tiivisteitä satunnaisluvuista 34
Taulukko 4. Muutamia laskennan historian merkkipaaluja. 45

Sisällys

1 JOHDANTO ... 1
1.1 Kvantti-informaatio ja klassinen informaatio ... 2
1.2 Kubittisääntöjä ... 2
1.3 Viritystila ja Paulin operaattorit ... 5
1.4 Lomittuminen ja hajoitettavuus ... 7

2 KVANTTIOHJELMISTOARKKITEHTUURIT, KVANTTILASKENTAAN LIITTYVÄT ALUSTAT ... 8
2.1 Q# ja QDK (Microsoft) ... 8
2.1.1 Groverin algoritmi ... 8
2.1.2 Deutsch–Jozsa-algoritmi ... 9
2.2 Qiskit ja Q Experience (IBM) ... 9
2.2.1 Bellin testi ... 10
2.2.2 Bernstein–Vazirani-algoritmi ... 12
2.3 Cirq (Google) ... 15
2.3.1 Kvanttiavainten jakaminen (QKD) ... 16
2.4 Muita ohjelmistoja, kieliä ja ohjelmistokehyksiä ... 18
2.5 Mikä on kvanttialijärjestelmien rooli ohjelmistoarkkitehtuureissa ja ohjelmistojärjestelmissä? ... 19

3 LASKENNALLINEN TUTKIMUS LIITTYEN AIHEESEEN KVANTTILASKENTA VS. KLASSINEN LASKENTA ... 21
3.1 Turingin koneeseen pohjautuva lähestyminen ... 21
3.2 Kvanttilaskennan odotetut hyödyt ja rajoitukset teoriasta ... 23
3.2.1 Shorin algoritmi ... 24

4 KAUPALLINEN PUOLI ... 26
4.1 Mitkä ovat odotetut roolit kvanttilaskennassa vs. klassisessa laskennassa 5 - 10 vuoden sisällä? ... 27
4.1.1 Reversiibelit tietokoneet ... 27
4.2 Mitkä ohjelmat / toimialueet kokevat suurimman vaikutuksen? ... 27
4.2.1 Satunnaislukujen generointi ... 27
4.2.2 Kvanttifysiikan ja kvanttikemian simulointi ... 30
4.2.3 Kvantinjälkeinen kryptografia ... 30
4.2.4 Ohjattu ja ohjaamaton oppiminen sekä kvanttivahvistusoppiminen kvanttitietokoneilla ... 37

5 LYHYT KUVAUS KVANTTITIETOKONELAITTEISTON KEHITYKSEN NYKYTILASTA JA TULEVAISUUDESTA ... 38
5.1 IBM ... 38
5.2 Google ... 40
5.3 Rigetti Computing ... 40
5.4 D-Wave Systems, Inc. ... 40

5.5 Kiinan tiede- ja teknologiayliopisto (USTC) ... 41
5.6 Xanadu ... 41
5.7 Honeywell ... 42
5.8 PsiQuantum ... 42
5.9 IonQ ... 42
5.10 VTT ja IQM ... 42
5.11 Kvanttilaitteistojen arvioidut kapasiteetit ja rajoitukset 5 - 10 vuoden sisällä ... 42

6 MUUN AIHEESEEN LIITTYVÄN TUTKIMUKSEN HAHMOTELMA ... 44
6.1 Analoginen laskenta ... 44
6.2 Ternäärinen laskenta ... 46
6.3 Reversiibeli laskenta ... 47
6.4 Uudet elektroniikan peruskomponentit: memristori, memkondensaattori, memkäämi ... 50
6.5 Klassinen, digitaalinen, binäärinen, ei-reversiibeli laskenta ... 50

7 YHTEENVETO ... 52

LÄHTEET ... 53

LIITTEET ... 65
A Liite: Kvanttilaskennan trendit vuodelle 2020 ja eteenpäin ... 65
A.1 Tekoäly, lohkoketjut ja kvanttilaskenta ... 65
A.2 Kvanttitietokone tekee jotain uutta ... 65
A.3 Kyberturvallisuudesta ... 65
B Liite: Kvanttialgoritmit ja kvanttisovellukset ... 66
B.1 Kyberturvallisuudesta ... 66
B.2 Kvanttifysiikan simulointi kvanttitietokoneella ... 67
C Liite: Kvanttilaskennan kurssit, tutkinnot ja tutkimusryhmät ... 67
C.1 Johtavia tutkimuslaitoksia kvanttilaskennassa ... 67
C.2 Kurssitarjontaa lähiopetuksena ja etäopetuksena ... 67
D Liite: Kvanttilaskenta ja digitaalinen turvallisuus ... 68
D.1 Kryptovaluutat turvaan kvanttitietokonehyökkäyksiltä ... 68
D.2 Suojautumista kvanttitietokonehyökkäyksiltä ... 69

1 Johdanto

1800-luvun loppupuolen fysiikka oli pääasiassa klassista mekaniikkaa, sähkömagnetismia ja termodynamiikkaa. Kvanttimekaniikka sai alkunsa, kun klassinen fysiikka ei enää kyennyt selittämään kaikkia havaittuja ilmiöitä 1800-luvun loppupuolella ja 1900-luvun alkupuolella. Klassista fysiikkaa haastoivat Albert Einsteinin vuonna 1905 esittämä suppea suhteellisuusteoria, joka osoitti, että Newtonin mekaniikan lait eivät päde nopeuksilla, jotka ovat lähellä tyhjiössä kulkevan valon nopeutta (299 792 458 m/s), sekä ydinfysiikka, joka mullisti käsityksemme atomin rakenteesta. (Hirvensalo 2003)

Vuonna 1900 Max Planck selitti mustan kappaleen säteilyn keksimällä uuden luonnonvakion, jota kutsutaan Planckin vakioksi

$$h = 6,63 \cdot 10^{-34}\,\mathrm{Js}. \tag{1.1}$$

Vuonna 1905 Albert Einstein selitti valosähköisen ilmiön siten, että valo käyttäytyy kuin kokoelma hiukkasia, joita kutsutaan fotoneiksi, joista itse kunkin energia on

$$E = hf \qquad , \tag{1.2}$$

missä f on säteilyn taajuus. (Hirvensalo 2003) Kvanttimekaniikassa, ja muussa kvanttiteoriaan liittyvässä, Planckin vakio (1.1) tulee toistuvasti vastaan.

Vuonna 1980 Paul Benioff (Benioff 1980) esitti ensimmäisenä idean kvanttilaskennasta kuvailemalla Turingin koneen kvanttimekaanisen mallin. Lisäksi samana vuonna venäläinen matemaatikko Yu I. Manin (Manin 1980) esitti idean hieman epämääräisessä muodossa. Vuonna 1982 Richard Feynman (Feynman 1982) osoitti, että R:n hiukkasen kvanttifysikaalista systeemiä ei voi simuloida tavallisilla tietokoneilla ilman eksponentiaalista hidastumista. R:n hiukkasen klassisen fysiikan systeemiä voi tosin simuloida pelkällä polynomisella hidastumisella. Feynman myös ehdotti, että tämän kvanttifysikaalisen systeemin simuloinnin hidastumisen voisi välttää käyttämällä tietokonetta, jonka laskenta perustuu kvanttifysiikkaan. Vuonna 1985 David Deutsch (Deutsch 1985) formaalisti määritteli universaalin kvanttitietokoneen universaalin Turingin koneen analogiana. Näistä ideoista voidaan katsoa kvanttitietokoneiden saaneen alkunsa. (Hirvensalo 2003)

1.1 Kvantti-informaatio ja klassinen informaatio

Klassinen eli “tavallinen” tietokone käsittelee klassista informaatiota. Kvanttitietokone käsittelee kvantti-informaatiota. Kvanttitietokoneita käsittelee yleistajuisesti suomennettu teos (Brown ja Pietiläinen 2001).

Klassisen informaation perusyksikkö on bitti, kun taas kvantti-informaation perusyksikkö on kubitti. Klassista informaatiota mitataan Shannonin entropialla, kun taas kvantti-informaatiota mitataan von Neumannin entropialla. (*Quantum information - Wikipedia*)

Klassiset digitaaliset tilat ovat diskreettejä, yleensä bittejä: 0 ja 1, tai joskus trittejä: esimerkiksi -1, 0 ja 1, kun taas kubitti on jatkuva-arvoinen - eräänlainen vektori Blochin pallolla. Vaikka kubitti on jatkuva-arvoinen, se on pienin mahdollinen yksikkö kvantti-informaatiota. Kvantti-informaation käsittelyä kuvaavat seuraavat teoreemat (*Quantum information - Wikipedia*)*:*

1. ei-teleportaatiota-teoreema, joka sanoo, että kubittia ei voi kokonaisuudessaan muuttaa klassiseksi bitiksi; eli sitä ei voi “lukea”.
2. ei-kloonaamista-teoreema, joka estää kopioiden ottamista mielivaltaisesta kubitista
3. ei-poistamista-teoreema, joka estää mielivaltaisen kubitin poistamista
4. ei-levittämistä-teoreema, joka estää yksittäisen kubitin lähettämisen useille vastaanottajille, vaikka sen voi siirtää paikasta toiseen (esimerkiksi kvanttiteleportaation välityksellä)
5. ei-piilottamista-teoreema, joka demonstroi kvantti-informaation säilymisen
6. ei-viestintää-teoreema (tai ei-merkinantoa-sääntö), joka sanoo, että lomittuneen kvanttitilan mittauksen aikana havainnoijan ei ole mahdollista välittää informaatiota toiselle havainnoijalle kokonaistilan alijärjestelmää mittaamalla.

1.2 Kubittisääntöjä

Kubitti on kvanttijärjestelmä, jossa on kaksi tasoa, joita merkitään Paul Diracin bra-ket-merkintätavan (’bracket’, sulkumerkki) mukaisesti $|0\rangle$ ja $|1\rangle$. Esimerkiksi, sisätulossa $\langle\phi|\psi\rangle$, bra-vektori on $\langle\phi|$ ja ket-vektori on $|\psi\rangle$.

1. Hilbertin avaruus on lineaarinen vektoriavaruus, jossa on kompleksiset kertoimet ja sisätulot

$$\langle \phi | \psi \rangle = \sum_i \phi_i^* \psi_i. \tag{1.3}$$

2. Yksittäiselle kubitille on kaksi standardia ortogonaalista tilaa (laskennallinen kanta / perustilat), joita merkitään

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \tag{1.4}$$

ja

$$|1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}. \tag{1.5}$$

3. Muita kubittitiloja ovat superpositiotilat

$$|+\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix} \quad , \tag{1.6}$$

$$|-\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ -1 \end{pmatrix} \quad , \tag{1.7}$$

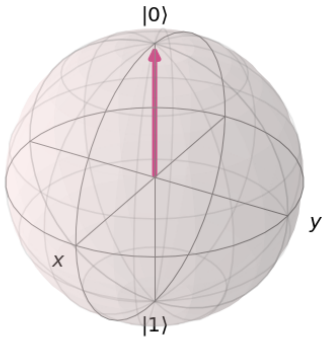
sekä

$$|\circlearrowright\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ i \end{pmatrix} \quad , \tag{1.8}$$

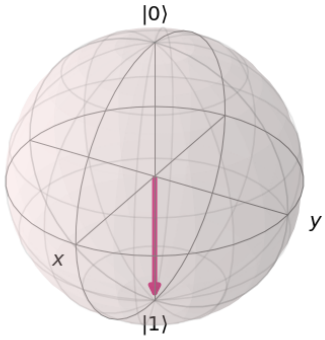
$$|\circlearrowleft\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ -i \end{pmatrix}. \tag{1.9}$$

4. Unitaarisuus tarkoittaa lineaarisuutta ja sisätulon säilyvyyttä.
5. Kahden kubitin järjestelmä voi olla tulotilassa kuten $|00\rangle$ tai $|0+\rangle$, mutta myös *lomittuneessa* tilassa $(|00\rangle + |11\rangle)/\sqrt{2}$, jossa kumpikaan kubitti ei ole tarkalleen määritellyssä tilassa, vaikka pari yhdessä onkin.
6. Mittaus saa järjestelmän käyttäytymään probabilisesti ja unohtamaan sen mittausta edeltäneen tilansa paitsi, jos kyseinen tila on sattumoisin kokonaan yhdessä aliavaruudessa Π_j.

Kubittitila $|0\rangle$ (1.4) on kuvattuna Blochin pallolla kuvassa 1. Kubittitila $|1\rangle$ (1.5) on kuvattuna Blochin pallolla kuvassa 2.



Kuvio 1. Kubittitila $|0\rangle$ kuvattuna Blochin pallolla vektorina.



Kuvio 2. Kubittitila $|1\rangle$ kuvattuna Blochin pallolla vektorina.

Kvanttilaskennan kaksi toimintaperiaatetta ovat

1. Fyysinen järjestelmä, joka on tarkassa tilassa, voi silti käyttäytyä satunnaisesti.
2. Kaksi järjestelmää, jotka ovat liian kaukana toisistaan vaikuttaakseen toisiinsa, voivat silti käyttäytyä, vaikkakin yksilöllisesti satunnaisesti, tavoilla, jotka ovat jotenkin voimakkaasti korreloituneet.

Toinen toimintaperiaate sisältää lomittumisen/kietoutumisen ajatuksen. Lomittuminen on kuitenkin enemmän kuin täydellinen klassinen korrelaatio. Jos kaksi henkilöä lukee saman paperin, he oppivat täysin saman informaation. Jos kolmaskin henkilö lukee saman paperin, hänkin oppii saman informaation. Kaikki kolme henkilöä ovat tässä tapauksessa täysin korreloituneita, ja he myös pysyvät korreloituneina, vaikka joutuisivat eroon toisistaan. Kvanttilomittuminen tarkoittaisi sitä, että jos kaksi henkilö lukee saman kvanttipaperin, he eivät tiedä mitä informaatiota oli kvanttipaperissa, ennen kuin he ovat yhdessä ja jakavat informaation.

Samalla he voivat myös vapauttaa kvanttipaperista enemmän informaatiota kuin alun perin näytti olevan mahdollista. (*Docs and Resources - IBM Quantum Experience - Entanglement*)

Käytetään tietotekniikasta tuttua merkintätapaa, jossa vähiten merkitsevä bitti (LSB) on oikealla. Tällöin kubittien q_0 ja q_1 välinen tensoritulo on $q_1 \otimes q_0$. Kontrolloitu EI (CNOT) on matriisimuodossa

$$CNOT = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \quad , \tag{1.10}$$

jossa q_0 on kontrollikubitti ja q_1 on kohdekubitti. Yhtälön (1.10) matriisi on itse asiassa kvanttiportti, jota käytetään kubittien operoimiseen kvanttitietokoneissa! Se muuttaa kohdekubitin nollasta ykköseksi tai ykkösestä nollaksi, jos kontrollikubittina on ykköskubitti. Jos kontrollikubittina on nollakubitti, kohdekubitille ei tapahdu mitään.

Toinen tärkeä kvanttiportti on Hadamardin portti, joka on matriisimuodossa

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}. \tag{1.11}$$

Hadamardin porttia (1.11) tarvitaan superpositiotilojen (1.6) ja (1.7) luomiseen!

Kahden kubitin lomittunut tila saadaan laittamalla H-portti kontrollikubitin eteen, ja sen jälkeen laittamalla CNOT-portti yhdistämään kubitteja. Tästä saadaan maksimaalisesti lomittunut kahden kubitin tila nimeltään Bellin tila. Lopuksi kubitit vielä mitataan, jolloin saadaan klassiseen rekisteriin kaksi bittiä. Kyseinen kvanttipiiri on kuvassa 3.

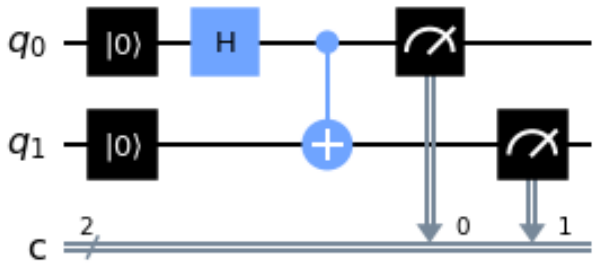
1.3 Viritystila ja Paulin operaattorit

Perustilalta (1.4) on mahdollista nousta viritystilalle (1.5) X-kvanttiportin eli X-kvanttioperaattorin (1.16) avulla siten, että

$$X|0\rangle = |1\rangle. \tag{1.12}$$

Vastaavasti

$$X|1\rangle = |0\rangle. \tag{1.13}$$



Kuvio 3. Kvanttipiiri, joka toteuttaa Bellin tilan zz-mittaukset.

X-kvanttiportti vastaakin klassisen tietokoneen EI-porttia!

Yhteen kubittiin operoiva kvanttiportti on matemaattisesti unitaarinen 2×2 matriisi. Matemaattisesti esitettynä

$$|\psi'\rangle = U|\psi\rangle \qquad \text{, missä} \tag{1.14}$$

$U^\dagger U = 1$.

Paulin operaattoreilla eli Paulin matriiseilla tarkoitetaan matriiseja

$$I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \qquad , \tag{1.15}$$

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \qquad , \tag{1.16}$$

$$Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \tag{1.17}$$

sekä

$$Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} . \tag{1.18}$$

On lukuisia muitakin tärkeitä kvanttiportteja kuten S, $S^\dagger$, T, $T^\dagger$, $R_Z(\theta)$ ja Toffolin portti (kontrolloitu-kontrolloitu EI, CCNOT). Näistä, ja muista kvanttilaskentaan liittyvistä matemaattisista konsepteista, kerrotaan tarkemmin lukuisissa kvanttilaskentaa käsittelevissä kirjoissa (Metodi, Faruque ja Chong 2011; Yanofsky ja Mannucci 2008; Rieffel ja Polak 2014;

S. Aaronson 2013; Moran 2019; Krishnakumar 2020; Sutor 2019; Norlén 2020; Loredo 2020; Silva 2018; Hidary 2019; Johnston, Harrigan ja Gimeno-Segovia 2019; Kaiser ja Granade 2021; Vos 2021).

1.4 Lomittuminen ja hajoitettavuus

Kahden kubitin järjestelmä on neliulotteinen Hilbertin avaruus $H_4 = H_2 \otimes H_2$, jonka ortonormaalina kantana on $\{|0\rangle|0\rangle, |0\rangle|1\rangle, |1\rangle|0\rangle, |1\rangle|1\rangle\}$. On tapana yksinkertaistaa notaatiota tyyliin $|0\rangle|0\rangle = |00\rangle$, $|0\rangle|1\rangle = |01\rangle$, ja niin edelleen. Kahden kubitin järjestelmän tila on yksikköpituinen vektori

$$c_0|00\rangle + c_1|01\rangle + c_2|10\rangle + c_3|11\rangle \qquad , \tag{1.19}$$

jolle $|c_0|^2 + |c_1|^2 + |c_2|^2 + |c_3|^2 = 1$. (Hirvensalo 2003)

Kahden kubitin tila $z \in H_4$ on hajoitettava, jos z voidaan kirjoittaa H_2:n tilojen tulona $z = x \otimes y$. Tila, joka ei ole hajoitettava, on lomittunut. Esimerkiksi, tila $\frac{1}{2}(|00\rangle + |01\rangle + |10\rangle + |11\rangle)$ on hajoitettava, sillä

$$\frac{1}{2}(|0\rangle|0\rangle + |0\rangle|1\rangle + |1\rangle|0\rangle + |1\rangle|1\rangle) = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \tag{1.20}$$

Toisaalta, tila $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$ on lomittunut. Oletetaan päinvastoin, että

$$\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) = (a_0|0\rangle + a_1|1\rangle)(b_0|0\rangle + b_1|1\rangle) = a_0b_0|00\rangle + a_0b_1|01\rangle + a_1b_0|10\rangle + a_1b_1|11\rangle \tag{1.21}$$

joillekin kompleksiluvuille a_0, a_1, b_0 ja b_1. Samalla saadaan seuraavat yhtälöt

$$\begin{cases} a_0b_0 = \dfrac{1}{\sqrt{2}} \\ a_0b_1 = 0 \\ a_1b_0 = 0 \\ a_1b_1 = \dfrac{1}{\sqrt{2}} \end{cases} \qquad , \tag{1.22}$$

jotka eivät voi pitää paikkaansa. Siispä tila $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$ on lomittunut. (Hirvensalo 2003)

2 Kvanttiohjelmistoarkkitehtuurit, kvanttilaskentaan liittyvät alustat

2.1 Q# ja QDK (Microsoft)

Microsoftin avoimen lähdekoodin ohjelmointikieli kvanttialgoritmien kehittämiseen ja ajamiseen on nimeltään Q#. QDK puolestaan on Quantum Development Kit, joka sisältää Q#-kirjastoja, kvanttisimulaattoreita, laajennuksia muille ohjelmointiympäristöille ja API-dokumentaation. QDK sisältää lisäksi kemiaan, koneoppimiseen ja numeriikkaan liittyviä kirjastoja. Ohjelmointikielenä Q# muistuttaa paljon Pythonia, C#:ia ja F#:ia. Q#-ohjelmia voi ajaa konsolisovelluksena, Jupyter Notebookin kautta tai käyttäen isäntänä Python- ja .NET-kieliä. (*What are the Q programming language and QDK?*)

2.1.1 Groverin algoritmi

Groverin etsintäalgoritmi on yksi suosituimmista kvanttilaskenta-algoritmeista, ja se on mahdollista toteuttaa Q#:llä. Esimerkki osoittaa, että simulaatio etsii tietyn merkkijonon epäjärjestyksessä olevasta listasta murto-osassa siitä ajasta, joka koko listan etsintään menisi klassisella tietokoneella. Groverin algoritmi vastaa esimerkiksi kysymykseen: Onko tämä pakasta vedetty kortti herttaässä? (*Run Grover's search algorithm in Q - Quantum Development Kit - Microsoft Quantum | Microsoft Docs*)

Groverin etsintäalgoritmilla kvanttitietokone todistetusti käy etsinnän läpi vähemmillä askelilla kuin mitä listassa on kohteita. Tähän ei klassinen algoritmi kykene. Groverin algoritmi kysyy, onko listassa oleva kohde etsimämme. Ensin järjestämättömän listan kohteet indeksoidaan. Sitten luodaan kvanttisuperpositio listan indekseistä, ja kukin kerroin, tai todennäköisyysamplitudi, edustaa todennäköisyyttä sille, että kyseinen indeksi on etsimämme. Algoritmissa on kaksi askelta, jotka vähitellen lisäävät sitä kerrointa, jota haemme, kunnes kyseisen kertoimen todennäköisyysamplitudi on lähellä arvoa yksi. Vähittäisten lisäysten määrä on pienempi kuin listassa olevien kohteiden määrä. Tämän takia Groverin etsintäalgoritmi suoriutuu etsinnästä vähemmin askelin kuin mikään klassinen algoritmi. (*Run Grover's search*

algorithm in Q - Quantum Development Kit - Microsoft Quantum | Microsoft Docs)

Osoitteesta https://web.archive.org/web/20210108142834/https://docs.microsoft.com/en-us/quantum/tutorials/search löytyy esimerkki Groverin algoritmista toteutettuna Q#-kielellä.

2.1.2 Deutsch–Jozsa-algoritmi

Deutsch–Jozsa-algoritmi (Deutsch ja Jozsa 1992) näytti ensimmäisenä eron ongelman kvanttivaikeuden ja klassisen vaikeuden välillä. Tässä algoritmissa on merkitystä sillä, että kvanttiamplitudit voivat olla sekä positiivisia että negatiivisia, kun taas klassiset todennäköisyydet ovat aina ei-negatiivisia. (*Docs and Resources - IBM Quantum Experience - Deutsch-Jozsa algorithm*)

Millaiseen ongelmaan tämä algoritmi sopii? Ajatellaan funktiota $f(x)$, joka ottaa syötteenä n-bittisiä tekstijonoja x ja palauttaa joko arvon 0 tai 1. $f(x)$ on joko vakiofunktio, joka saa saman arvon $c \in \{0,1\}$ kaikilla syötteillä x tai tasapainotettu funktio, joka saa jokaisen arvon 0 ja 1 tasan puoliksi eri syötteille. Tavoitteena on arvioida, onko f vakio vai tasapainotettu tekemällä mahdollisimman vähän funktioarviointeja. Klassisesti tarvitaan $2^{n-1}+1$ funktioarviointia pahimmassa tapauksessa. Deutsch–Jozsa-algoritmilla kysymykseen löytyy vastaus vain yhdellä funktioarvioinnilla. (*Docs and Resources - IBM Quantum Experience - Deutsch-Jozsa algorithm; Deutsch-Jozsa Algorithm - A university quantum algorithms/computation course supplement based on Qiskit*)

Osoitteesta https://web.archive.org/web/20210112234327/https://raw.githubusercontent.com/oreilly-qc/oreilly-qc.github.io/master/samples/QSharp/ch14_DJ_deutsch_jozsa.qs löytyy esimerkki Deutsch–Jozsa-algoritmista toteutettuna Q#-kielellä.

2.2 Qiskit ja Q Experience (IBM)

Kvanttilaskennan teho perustuu kahteen kvanttimekaniikan kulmakiveen eli superpositioon ja lomittumiseen. Qiskit on ohjelmistokehityspaketti (SDK), joka hyödyntää näitä kvanttimekaanisia sääntöjä käyttäen kielenään kvanttipiirejä. Kvanttipiirit, jotka koostuvat kvanttiporteista, käskyistä ja klassisesta kontrollilogiikasta, mahdollistavat monimutkaisten algorit-

mien ja sovellusten ajamisen kvanttitietokoneessa. Qiskit on pohjimmiltaan moottori kvanttipiirien tekemiseen, optimointiin ja ajamiseen. Ylimääräiset algoritmi- ja sovelluskerrokset hyödyntävät kvanttipiirejä, usein yhdessä klassisten laskentaresurssien kanssa, ratkaistakseen ongelmia optimoinnissa, kvanttikemiassa, fysiikassa, koneoppimisessa ja taloudessa. (*Quantum computing in a nutshell - Qiskit 0.23.2 documentation*)

Q Experience on nettiselaimella toimiva kvanttitietokonealusta, jonka avulla voi käyttää IBM:n pilvipalvelun kautta oikeita kvanttitietokoneita ja myös kvanttitietokonesimulaattoria.

2.2.1 Bellin testi

Bellin testi osoittaa, että lomittuneen tilan mittaukset eivät selity lokaalilla piilomuuttujateorialla, joten täytyy olla korrelaatioita, jotka ovat klassisen teorian ulottumattomissa. Bellin tila tai EPR-pari on

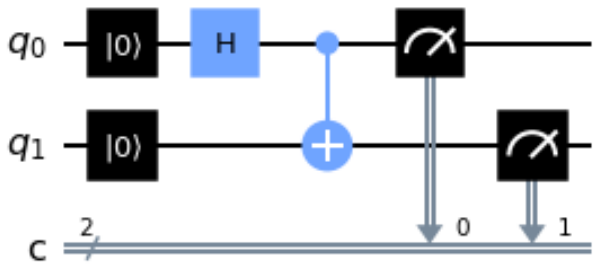
$$\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle). \tag{2.1}$$

Tämä lomittunut tila on mielenkiintoinen, koska yhden kubitin mittaaminen romahduttaa superposition ja kertoo myös toisen kubitin tilan. Esimerkiksi, oheinen mittaus on mahdollinen

$$\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \quad \xrightarrow{\text{mittaus}} \quad |11\rangle. \tag{2.2}$$

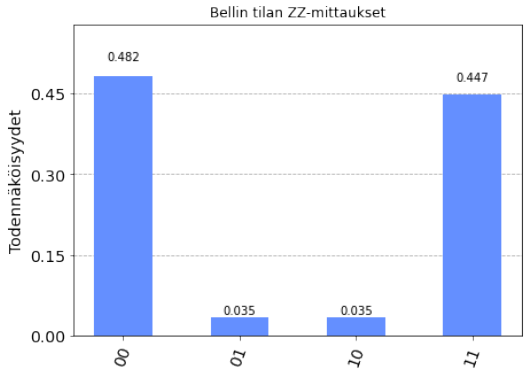
Vaikka nämä kubitit olisivat valovuosien päässä toisistaan, yhden kubitin mittaaminen romahduttaa superposition ja vaikuttaa välittömästi toiseen. Tätä “aavemaista kaukovaikutusta” varhaisen 1900-luvun fyysikot kummastelivat suuresti. (*Multiple Qubits and Entangled States - A university quantum algorithms/computation course supplement based on Qiskit*) Nykyään vielä kiistellään paljon kvanttimekaniikan tulkinnoista. Erityisen mielenkiintoinen on monimaailmatulkinta, jonka mukaan on olemassa rinnakkaismaailmankaikkeuksia, jotka ovat kuitenkin jollain tavalla rinnakkain samassa avaruudessa ja ajassa oman maailmamme kanssa. Muiden maailmojen olemassaolo poistaisi aaltofunktion romahtamisen, satunnaisuuden ja kaukovaikutuksen kvanttiteoriasta ja kaikesta fysiikasta. Monimaailmatulkintaa käsittelevät Stanfordin yliopiston sivu (*Many-Worlds Interpretation of Quantum Mechanics*), Princetonin fysiikkasarjan kirja (DeWitt, Everett ja Graham 1973) sekä yleistajuiseen muotoon kirjoitettu kirja (Brown ja Pietiläinen 2001).

Qiskitillä voidaan luoda kvanttipiiri, joka selvittää, mitä tapahtuu Bellin tilassa standardikannassa mitattaessa. Kvanttipiiri on kuvassa 4. Histogrammi kuvassa 5 kertoo, että tilat



Kuvio 4. Kvanttipiiri, joka toteuttaa Bellin tilan ja tekee zz-mittaukset kubiteille.

$|00\rangle$ ja $|11\rangle$ ovat lähes yhtä todennäköisiä. Teorian mukaan kummankin tilan todennäköisyys on 50%. Lisäksi histogrammiin on päätynyt myös tiloja $|01\rangle$ ja $|10\rangle$, vaikka teoria ei niitä ennusta. *Nämä tilat ovat kvanttilaskentaan liittyviä virheitä! Tulevaisuuden kvanttitietokoneissa näiden virheellisten tilojen määrän odotetaan vähenevän.*



Kuvio 5. Bellin tilan zz-mittausten histogrammista selviää eri tilojen todennäköisyydet oikeassa kvanttitietokoneessa ajettuna.

```
%matplotlib inline
from qiskit import QuantumRegister, ClassicalRegister, QuantumCircuit,
    execute, Aer
from qiskit.visualization import plot_histogram, plot_bloch_vector
from numpy import pi
from math import sqrt
```

```
qreg_q = QuantumRegister(2, 'q')
creg_c = ClassicalRegister(2, 'c')
circuit = QuantumCircuit(qreg_q, creg_c)
circuit.reset(qreg_q[0])
circuit.h(qreg_q[0])
circuit.reset(qreg_q[1])
circuit.cx(qreg_q[0], qreg_q[1])
circuit.measure(qreg_q[0], creg_c[0])
circuit.measure(qreg_q[1], creg_c[1])
circuit.draw(output='mpl')
from qiskit import IBMQ
IBMQ.load_account()
provider = IBMQ.get_provider('ibm-q')
qcomp = provider.get_backend('ibmq_16_melbourne')
from qiskit.tools.monitor import job_monitor
job = execute(circuit, backend=qcomp)
result = job.result()
job_monitor(job)
plot_histogram(result.get_counts(circuit))
```

Listing 2.1. Bellin tilan mittauksia Qiskitillä.

2.2.2 Bernstein–Vazirani-algoritmi

Bernstein–Vazirani-algoritmi (Bernstein ja Vazirani 1997) on laajennus Deutsch–Jozsa-algoritmille. Oletetaan, että laatikossa on piilossa tuntematon kuusibittinen luku. Arvaamalla voidaan pahimmassa tapauksessa tarvita $2^6 = 64$ yritystä luvun löytämiseksi. Klassinen tietokone tarvitsee kuusi yritystä selvittääkseen, mikä 6-bittinen luku on kyseessä. Kvanttitietokone voi selvittää luvun yhdellä yrityksellä. (*Bernstein-Vazirani Algorithm — Programming on Quantum Computers Season 1 Ep 6; Bernstein-Vazirani Algorithm - A university quantum algorithms/computation course supplement based on Qiskit*)

Vastaavasti n-bittiselle luvulle arvauksia tarvitaan enintään 2^n kappaletta. Klassinen tietokone tarvitsee tähän n kappaletta yrityksiä. Kvanttitietokoneelle riittää jälleen vain yksi yritys. (*Bernstein-Vazirani Algorithm — Programming on Quantum Computers Season 1 Ep 6; Bernstein-Vazirani Algorithm - A university quantum algorithms/computation course supple-*

ment based on Qiskit)

Klassinen tietokone etsii 6-bittisen salaisen luvun 110010 esimerkiksi seuraavanlaisesti JA-portteja hyödyntämällä: Algoritmin askeleet näkyvät taulukossa 1. Salainen luku löytyy, kun luetaan taulukon viimeinen sarake alhaalta ylöspäin: 110010. (*Bernstein-Vazirani Algorithm — Programming on Quantum Computers Season 1 Ep 6; Bernstein-Vazirani Algorithm - A university quantum algorithms/computation course supplement based on Qiskit*)

salaisen luvun bitti (JA-portin syöte A)	kokeiltava bitti (JA-portin syöte B)	arvaus (JA-portin tuloste C)
110010	000001	0
110010	000010	1
110010	000100	0
110010	001000	0
110010	010000	1
110010	100000	1

Taulukko 1. Klassisen algoritmin askeleet salaisen luvun arvaamiseksi.

```
# L hde: "Bernstein-Vazirani Algorithm     Programming on Quantum
    Computers Season 1 Ep 6" <URL:https://www.youtube.com/watch?v=
    sqJIpHYl7oo>
%matplotlib inline
from qiskit import *
from qiskit.tools.visualization import plot_histogram

secretnumber = '110010'

circuit = QuantumCircuit(len(secretnumber)+1, len(secretnumber))

circuit.h(range(len(secretnumber)))
circuit.x(len(secretnumber))
circuit.h(len(secretnumber))

circuit.barrier()

```

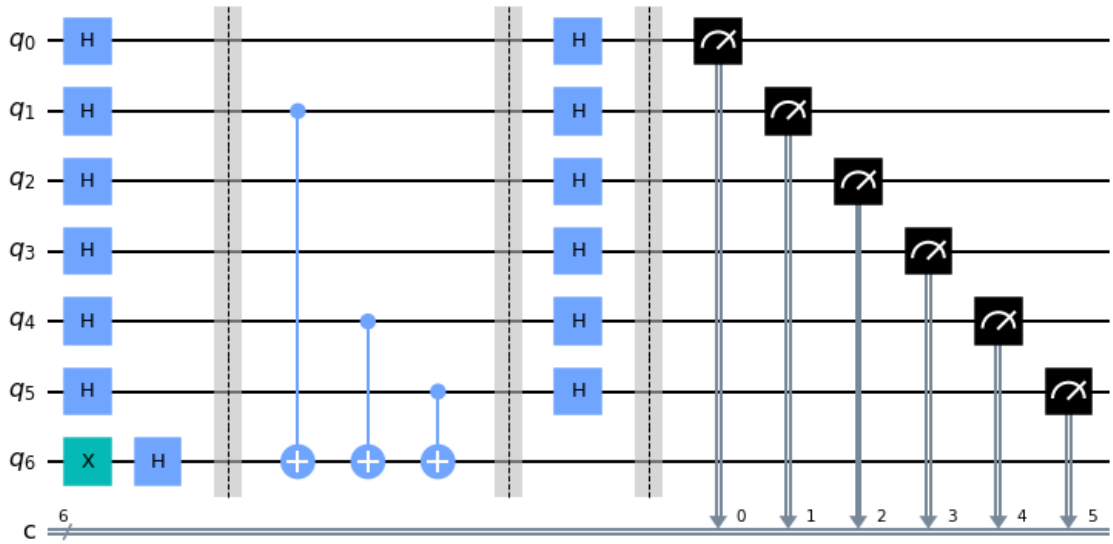
```
for ii, yesno in enumerate(reversed(secretnumber)):
    if yesno == '1':
     circuit.cx(ii, len(secretnumber))

circuit.barrier()
circuit.h(range(len(secretnumber)))
circuit.barrier()
circuit.measure(range(len(secretnumber)), range(len(secretnumber)))

simulator = Aer.get_backend('qasm_simulator')
result = execute(circuit, backend = simulator, shots = 1).result()
counts = result.get_counts()
print(counts)
circuit.draw("mpl")
```

Listing 2.2. Bernstein–Vazirani-algoritmi Qiskitillä.

Qiskitillä voidaan luoda Bernstein–Vazirani-algoritmin toteuttava kvanttipiiri, joka etsii 6-bittisen salaisen luvun, tulostaa ruudulle kvanttipiirin (Kuva 6) ja itse salaisen luvun: 110010.



Kuvio 6. Kvanttipiiri, joka toteuttaa Bernstein–Vazirani-algoritmin.

2.3 Cirq (Google)

Cirq on avoimen lähdekoodin ohjelmistokehys NISQ-tietokoneille (engl. “Noisy Intermediate Scale Quantum computers”). Sen on kehittänyt Googlen AI Quantum Team, ja se esiteltiin “International Workshop on Quantum Software and Quantum Machine Learning”-työpajassa 18.7.2018. (*Cirq - Wikipedia*)

Ohessa on esimerkki, miten cirq:lla luodaan kutritti (kolmiarvoista logiikkaa käyttävän tritin kvanttimekaaninen vastine).

```
# L hde: "Qudits | Cirq | Google Quantum AI" <URL:https://web.archive.
    org/web/20210122190459/https://quantumai.google/cirq/qudits>
%matplotlib inline
import cirq

class QutritPlusGate(cirq.SingleQubitGate):
    def _qid_shape_(self):
        return (3,)

    def _unitary_(self):
        return np.array([[0, 0, 1],
                         [1, 0, 0],
                         [0, 1, 0]])

    def _circuit_diagram_info_(self, args):
        return '[+1]'

q0 = cirq.LineQid(0, dimension=3)
circuit = cirq.Circuit(
    QutritPlusGate().on(q0)
)
print(circuit)
```

Listing 2.3. Kutritin luominen cirq:lla.

2.3.1 Kvanttiavainten jakaminen (QKD)

"BB84 on kvanttiavainten jakamisprotokolla (QKD), jonka kehittivät Charles Bennett ja Gilles Brassard vuonna 1984. Se oli ensimmäinen kvanttikryptografiaprotokolla, joka käytti kvanttimekaniikan lakeja kuten ei-kloonaamista, todistetusti turvallisten avainten luontiin. Sen toiminta perustuu siihen, että on mahdotonta saada informaatiota tunnistamaan kahta ei-ortogonaalista tilaa ilman, että signaali häiriintyy." (*Example program to demonstrate BB84 QKD Protocol*)

"Alicen ja Bobin välillä on klassinen kommunikointikanava. Lisäksi Alice voi valmistaa kubitteja tiettyyn tilaan ja lähettää niitä Bobille käyttäen yksisuuntaista kvanttikanavaa... Alice luo kaksi satunnaista binäärimerkkijonoa a ja b, joiden pituus on n. Merkkijono a koodaa tilan ja merkkijono b koodaa kannan. Sen jälkeen Alice valmistelee n kubittia seuraavasti

$$|q_i\rangle = |0\rangle, \text{ jos } a_i = 0,\ b_i = 0; \tag{2.3}$$

$$|q_i\rangle = |1\rangle, \text{ jos } a_i = 1,\ b_i = 0; \tag{2.4}$$

$$|q_i\rangle = |+\rangle, \text{ jos } a_i = 0,\ b_i = 1; \tag{2.5}$$

ja

$$|q_i\rangle = |-\rangle, \text{ jos } a_i = 1,\ b_i = 1. \tag{2.6}$$

(...) Alice lähettää hänen kubittinsa Bobille. Bob luo satunnaisen binäärimerkkijonon c, jonka pituus on n. Hän mittaa kubitin $|q_i\rangle$ laskennallisessa kannassa $\{|0\rangle, |1\rangle\}$, jos $c_i = 0$ ja Hadamardin kannassa (superpositiokannassa) $\{|+\rangle, |-\rangle\}$, jos $c_i = 1$ ja tallettaa tuloksen merkkijonoon m. Alice ja Bob ilmoittavat sitten merkkijonot b ja c, jotka enkoodaavat Alicen ja Bobin satunnaiset kantavalinnat." (*Example program to demonstrate BB84 QKD Protocol*)

"Merkkijonot a ja m täsmäävät kohdissa, joissa b ja c ovat samat. Tämä tapahtuu, koska tila mitattiin samassa kanssa kuin se valmisteltiin. Jäljelle jääville biteille tulokset ovat korreloimattomia. Bitit merkkijonoista a ja m, joissa kannat täsmäävät, voidaan käyttää kryptografian avaimena... BB84 on turvallinen sieppaa-ja-uudelleenlähetä-hyökkäyksiä vastaan. Ei-kloonaamista-teoreema takaa, että kubittia, joka on lähtökohtaisesti tunnistamattomassa tilassa, ei voida kopioida eikä kloonata. Siten, mikä tahansa mittaus tuhoaa kubitin alkutilan. Oletetaan, että salakuuntelija Eve sieppaa kaikki Alicen kubitit, mittaa ne satunnaisesti

valitussa kannassa, valmistelee toisen kubitin tilaan, jonka hän mittasi ja uudelleenlähettää sen Bobille. Tila, jonka Eve mittaa ei ole välttämättä tila, jonka Alice valmisteli, ja täten, Alice ja Bob eivät mittaa samoja tuloksia kyseiselle kubitille, vaikka heidän kantavalintansa täsmäisivät. Siten, Alice ja Bob voivat havaita salakuuntelun vertaamalla muutamia bittejä saamistaan avaimista." (*Example program to demonstrate BB84 QKD Protocol*)

Osoitteesta https://web.archive.org/web/20210108144928/https://raw.githubusercontent.com/quantumlib/Cirq/master/examples/bb84.py löytyy esimerkki BB84-protokollaa demonstroivasta ohjelmasta toteutettuna Python-kielen Cirq-kirjastolla.

```
Simuloidaan protokolla ilman salakuuntelijaa

0: ___X___M___________

1: ___H___H___M_______

2: ___X___H___M_______

3: ___X___H___M_______

4: ___X___H___M_______

5: ___X___H___H___M___

6: ___H___M___________

7: ___H___H___M_______

Alicen kanta:       CHCCCHCH
Bobin kanta:        CHHHHHHH
Alicen bitit:       10111100
Kannat tasmaavat:   XX___X_X
Odotettu avain:     1010
Todellinen avain:   1010
```

Listing 2.4. Simuloidaan protokolla ilman salakuuntelijaa.

```
Simuloidaan protokolla salakuuntelijan kanssa
```

```

0: ___H___M____________H___M____________

1: ___H___M____________H___H___M________

2: ___X___H___H___M___X___H___H___M___

3: ___H___M____________H___M____________

4: ___M________________M________________

5: ___X___H___M________X___H___M________

6: ___H___M____________X___H___M________

7: ___X___H___H___M___X___H___M________

Alicen kanta:       HCHCCHCH
Bobin kanta:        HHHCCHCC
Alicen bitit:       00100101
Kannat tasmaavat:   X_XXXXX_
Odotettu avain:     010010
Todellinen avain:   111011
```

Listing 2.5. Simuloidaan protokolla salakuuntelijan kanssa.

2.4 Muita ohjelmistoja, kieliä ja ohjelmistokehyksiä

Tässä raportissa on käsitelty ohjelmistoja ja kieliä kuten Q#, Qiskit ja Cirq. On lukuisia muitakin kvanttilaskentaan liittyviä ohjelmistoja, kieliä tai ohjelmistokehyksiä kuten

- D-Wave Leap (D-Wave): https://www.dwavesys.com/take-leap
- Forest SDK (Rigetti): https://pyquil-docs.rigetti.com/en/stable/
- Language-Integrated Quantum Operations, LIQUi|> (Microsoft): https://www.microsoft.com/en-us/research/project/language-integrated-quantum-operations-liqui/
- NumPy: https://numpy.org/

- OpenQASM (IBM): https://github.com/qiskit/openqasm
- Paddle Quantum (Baidu): https://github.com/PaddlePaddle/Paddle
- Pennylane (Xanadu): https://xanadu.ai/pennylane
- ProjectQ: https://projectq.ch/
- PyTorch: https://pytorch.org/
- QCEngine (O'Reilly Media): https://oreilly-qc.github.io/
- QCL: http://tph.tuwien.ac.at/~oemer/qcl.html
- Quantum Inspire (QuTech): https://www.quantum-inspire.com/
- Quipper: https://www.mathstat.dal.ca/~selinger/quipper/
- QuTiP: http://qutip.org/
- QWIRE: https://github.com/inQWIRE/QWIRE
- Silq: https://silq.ethz.ch/
- Strawberry Fields (Killoran ym. 2019): https://strawberryfields.ai/
- TensorFlow: https://www.tensorflow.org/

2.5 Mikä on kvanttialijärjestelmien rooli ohjelmistoarkkitehtuureissa ja ohjelmistojärjestelmissä?

Kvanttilaskenta tuo paljon laskentatehoa, mutta valitettavasti mukana tulee suuria rajoituksia. Kvanttilaskennan on oltava reversiibeliä: jos funktiosta tulee ulos tuloste B syötteellä A, on oltava keino saada A, kun syötetään käänteisfunktiolle B. Seuraus on, että jokaisella funktiolla on oltava yhtä monta syötettä ja tulostetta. Otetaan esimerkkinä JA-portti. Siinä on kaksi syötettä, mutta vain yksi tuloste. JA-portin tuloste on 1, jos ja vain jos molemmat syötteet ovat 1. JA-portin tuloste on 0, jos jompikumpi tai molemmat syötteistä on 0. Nollatulosteesta itsestään ei siis voi päätellä, mitkä JA-portin syötteet ovat olleet kyseessä. JA-portti ei siis ole reversiibeli. Reversiibeliys vaatii riittävästi informaation säilyttämistä, jotta laskennan välivaiheet voidaan palauttaa takaisin. (Hayes 2014)

Toinen suuri rajoitus kvanttilaskennassa on kubittien kopioinnin mahdottomuus, joka johtuu kvanttimekaniikan ei-kloonaamista-teoreemasta. Superpositiotila nimittäin romahtaa, kun tila mitataan. (Hayes 2014)

Edellisistä rajoituksista voidaan päätellä, että kvanttiohjelmien tulee noudattaa eräänlaista savupiippuarkkitehtuuria (engl. 'stovepipe architecture'), jossa informaatio virtaa suoraan läpi reversiibelien kvanttilogiikkaporttien päästä päähän. Aivan erityisesti kvanttiohjelmassa ei voi olla silmukoita, jossa hypätään ohjelman aikaisempaan kohtaan suorittamaan samat komennot uudelleen. (Hayes 2014)

On mahdollista, että kvanttitietokoneet eivät suoraan korvaa klassisia tietokoneita. Ohjeena onkin: "Älä rakenna puhdasta kvanttitietokonetta. Rakenna klassinen tietokone, jossa on mukana kvanttialijärjestelmä. Sen jälkeen luo tarvittavat ohjelmistot molemmille järjestelmille…" (Hayes 2014)

3 Laskennallinen tutkimus liittyen aiheeseen kvanttilaskenta vs. klassinen laskenta

Vahva Churchin–Turingin teesi sanoo, että “mikä tahansa laskennan malli on simuloitavissa probabilistisella Turingin koneella siten, että vaaditaan korkeintaan polynominen kasvu elementaaristen opraatioiden lukumäärälle.” Tämä on hyvä uutinen laskennallisen kompleksisuuden teorialle, koska siitä seuraa, että huomio voidaan kiinnittää probabilistisen Turingin koneen laskentamallille. (Nielsen ja Chuang 2010). Probabilistinen Turingin kone on määritelty kohdassa 3.1.2.

3.1 Turingin koneeseen pohjautuva lähestyminen

Nykyään käytössä olevat tietokoneet ovat enimmäkseen sähköisiä, binäärisiä, ei-reversiibeleitä, digitaalisia ja klassisia. Olemme niin tottuneita sellaisiin tietokoneisiin, että on vaikea uskoa, että tietokoneet voivat olla myös esimerkiksi mekaanisia, ternäärisiä, reversiibeleitä, analogisia tai kvanttimaisia!

Nykyään yleisesti käytössä olevat tietokoneet pohjautuvat Alan Turingin ja John von Neumannin luomiin malleihin. Tyypillisesti informaatio käsitellään binäärisessä muodossa, jossa on käytössä kaksi numeroa 0 ja 1. Tiedon perusosanen on siis tällaisessa järjestelmässä bitti. Bittejä käsitellään George Boolen algebraan pohjautuvien logiikkaporttien JA (AND), TAI (OR), EI (NOT), EI-JA (NAND), EI-TAI (NOR), ERI (XOR), SAMA (XNOR) avulla.

Universaalit kvanttitietokoneet käyttävät kvanttilogiikkaportteja, jotka operoivat pieneen määrään kubitteja. Kvanttilogiikkaportit ovat reversiibeleitä toisin kuin useimmat klassiset logiikkaportit! Klassista laskentaa voi kuitenkin suorittaa myös reversiibeleillä porteilla. Toffolin portti on esimerkki reversiibelistä portista, jolla voi toteuttaa kaikki Boolen funktiot. Toffolin portti tarvitsee usein apubitin (engl. ’ancilla bit’), joka on siis tavallaan ylimääräinen bitti. Toffolin portista on olemassa suora kvanttivastineensa, joka osoittaa, että kaikki klassisten piirien suorittamat operaatiot voidaan suorittaa kvanttipiireillä!

Tämän osion loppupuoli perustuu Mika Hirvensalon kvanttilaskentakirjaan (Hirvensalo 2003).

On tärkeää määritellä, mikä itse asiassa on laskukone/tietokone.

Määritelmä 3.1.1 *Deterministinen Turingin kone M yli aakkosten A on kuusimonikko (engl. "six-tuple")* $(Q, A, \delta, q_0, q_a, q_r)$*, missä A on äärellinen joukko perusinformaatioyksiköitä, Q on äärellinen joukko sisäisiä kontrollitiloja,* $q_0, q_a, q_r \in Q$ *ovat alkutila, hyväksyvä tila ja hylkäävä tila, ja siirtymäfunktio*

$$\delta : Q \times A \to Q \times A \times \{-1, 0, 1\}. \tag{3.1}$$

Turingin kone 3.1.1 kuvaa laskentaprosessin klassista mallia. Yleisesti uskotaan, että Turingin koneet käsittävät myös algoritmisen laskettavuuden: mikä on laskettavissa algoritmisesti, on laskettavissa myös Turingin koneella. (Hirvensalo 2003)

Deterministisen Turingin koneen määritelmää voidaan muokata sopivalla tavalla, jolloin saadaan malli probabilistiselle laskennalle. Tarvittava muokkaus on korvata siirtymäfunktio siirtymätodennäköisyysdistribuutiolla. (Hirvensalo 2003)

Määritelmä 3.1.2 *Probabilistinen Turingin kone M yli aakkosten A on kuusimonikko* $(Q, A, \delta, q_0, q_a, q_r)$*, missä A on äärellinen joukko perusinformaatioyksiköitä, Q on äärellinen joukko sisäisiä kontrollitiloja,* $q_0, q_a, q_r \in Q$ *ovat alkutila, hyväksyvä tila ja hylkäävä tila, ja siirtymätodennäköisyysdistribuutio*

$$\delta : Q \times A \times Q \times A \times \{-1, 0, 1\} \to [0, 1]. \tag{3.2}$$

Vaaditaan kaikille $(q_1, a_1) \in Q \times A$

$$\sum_{(q_2, a_2, d) \in Q \times A \times \{-1,0,1\}} \delta(q_1, a_1, q_2, a_2, d) = 1.$$

Malli kvanttilaskennalle saadaan, kun otetaan suoraviivainen yleistys probabilistisesta Turingin koneesta, jossa todennäköisyydet on korvattu siirtymäamplitudeilla. (Hirvensalo 2003)

Määritelmä 3.1.3 *Kvanttimainen Turingin kone (QTM) yli aakkosten A on kuusimonikko* $(Q, A, \delta, q_0, q_a, q_r)$*, missä A on äärellinen joukko perusinformaatioyksiköitä, Q on äärellinen*

joukko sisäisiä kontrollitiloja, $q_0, q_a, q_r \in Q$ *ovat alkutila, hyväksyvä tila ja hylkäävä tila, ja siirtymäamplitudifunktion*

$$\delta : Q \times A \times Q \times A \times \{-1, 0, 1\} \to \mathbb{C} \tag{3.3}$$

on täytettävä ehto

$$\sum_{(q_2, a_2, d) \in Q \times A \times \{-1,0,1\}} |\delta(q_1, a_1, q_2, a_2, d)|^2 = 1$$

mille tahansa $(q_1, a_1) \in Q \times A$.

On pystytty osoittamaan, että mikä tahansa, joka on laskettavissa Turingin koneella, on laskettavissa myös reversiibelillä Turingin koneella, johon myös kvanttimainen Turingin kone lukeutuu. *Siispä, kvanttimaiset Turingin koneet ovat ainakin yhtä tehokkaita kuin tavalliset Turingin koneet.* (Hirvensalo 2003)

Pystyykö kvanttimaisilla Turingin koneilla tekemään jotain sellaista, johon tavallinen Turingin kone ei kykene? Ei, ja sama pätee myös probabilistisille Turingin koneille. Tavallista Turingin konetta voi käyttää simuloimaan kvanttimaista Turingin konetta muistamalla kaikki yhtä aikaa olemassa olevat konfiguraatiot superpositiossa amplitudeineen. (Hirvensalo 2003)

On kuitenkin uskottavaa, että kvanttimaisia Turingin koneita ei voi tehokkaasti simuloida tavallisilla eikä edes probabilistisilla Turingin koneilla. Tässäkin päädytään polynomiaikaiseen Shorin kvanttialgoritmiin, joka etsii luvun alkulukutekijät. Klassista vastinetta sille ei ole kuitenkaan löydetty, ei edes probabilistista vastinetta, vaikka vaivaa on nähty paljon. (Hirvensalo 2003)

3.2 Kvanttilaskennan odotetut hyödyt ja rajoitukset teoriasta

On mahdollista, että kvanttitietokoneet ovat joissain tapauksissa selvästi nopeampia kuin klassiset tietokoneet, mutta useimmissa tapauksissa ne olisivat ehkä vain vähän klassisia tietokoneita nopeampia. Tämä osio perustuu enimmäkseen Scientific American -lehden artikkeliin (Scott Aaronson 2008) kvanttilaskennasta.

3.2.1 Shorin algoritmi

Vuonna 1994 Peter Shor löysi kvanttialgoritmin (Shor 1994), joka dramaattisesti nopeuttaa käytännöllisen ongelman ratkaisun löytämistä. Shorin algoritmin avulla kvanttitietokone voi jakaa tekijöihinsä n-numeroisen luvun käyttäen askelmääriä, jotka kasvavat vain kuin noin n^2 eli polynomisessa ajassa. Klassisen tietokoneen parhaat algoritmit käyttävät samaan ongelmaan askelmääriä, jotka kasvavat eksponentiaalisesti. (Scott Aaronson 2008)

```
# L hde: "Textbook and S h o r s algorithms - Qiskit 0.23.2 documentation
    " <URL:https://web.archive.org/web/20210113235334/https://qiskit.org/
    documentation/tutorials/algorithms/09_textbook_algorithms.html>
import math
import numpy as np
from qiskit import BasicAer
from qiskit.aqua import QuantumInstance
from qiskit.aqua.algorithms import Shor
N = 15
shor = Shor(N)
backend = BasicAer.get_backend('qasm_simulator')
quantum_instance = QuantumInstance(backend, shots=1024)
result = shor.run(quantum_instance)
print(f"Lista t e k i j i s t  luvulle {N} laskettuna Shorin algoritmilla on
    {result['factors'][0]}.")
```

Listing 3.1. Shorin algoritmi Qiskitillä.

```
Lista t e k i j i s t  luvulle 15 laskettuna Shorin algoritmilla on [3, 5].
```

Listing 3.2. Shorin algoritmin tuloste.

Tietojenkäsittelytieteessä on tärkeää tuntea, kuinka nopeasti tarkasteltavan ongelman ratkaisemisen ajankäyttö kasvaa ongelman koon kasvaessa. Aika mitataan algoritmin tarvitsemissa askeleissa ratkaisun saavuttamiseksi. Esimerkiksi, kahden n-numeroisen luvun kertominen vaatii ajan, joka kasvaa kuin numeroiden neliö, n^2, joten sanotaan, että tarvittava aika on "n:n suhteen polynominen". Luvun jakaminen tekijöihin puolestaan on vaikeampaa, koska parhaillakin tunnetuilla metodeilla menee aika, joka kasvaa eksponentiaalisesti suhteessa numeroiden määrään. (Scott Aaronson 2008)

Ongelmat, jotka suurillakin n:n arvoilla, ratkeavat mielekkäässä ajassa ovat niitä, joille on olemassa algoritmi, joka tarvitsee askeleita, jotka kasvavat kuin n korotettuna kiinnitettyyn potenssiin kuten n, n^2 tai $n^{2,5}$. Näitä algoritmeja kutsutaan tehokkaiksi, ja ongelmat, jotka ovat ratkaistavissa tehokkailla algoritmeilla, kuuluvat kompleksisuusluokkaan P, joka tarkoittaa "polynomista aikaa". Esimerkkinä P-ongelmasta on tiekarttaan liittyvä kysymys: Onko jokainen kaupunki saavutettavissa kaikista muista kaupungeista? (Scott Aaronson 2008)

Toinen karttaan liittyvä ongelma koskee sitä, voidaanko jokainen naapurimaa värittää eri värillä, kun väreinä ovat ainoastaan punainen, sininen ja vihreä. NP tarkoittaa "epädeterminististä polynomista aikaa". NP-luokan ongelman ratkaisu tarvitsee polynomisen ajan (esimerkiksi n^2), vaikka itse ratkaisun löytäminen voi olla hankalaa. (Scott Aaronson 2008) Esimerkiksi bitcoin-louhinnassa on vaikeaa löytää lohko, jonka SHA256-tiiviste on pienempi (luku sisältää riittävän monta etunollaa) kuin Bitcoin-verkon asettama tavoite. Kun tällainen lohko löytyy, on sen tarkistaminen ja validointi helppoa. Bonneaun artikkelin mukaan (*Bitcoin mining is NP-hard*) bitcoin-louhinta on sekä NP-kovaa että NP:tä, ja samalla siis myös NP-täydellistä.

NP-ongelmiin sisältyy suuri joukko kiinnostavia ongelmia. Kaikki P-ongelmat ovat samalla myös NP-ongelmia. NP-täydelliset ongelmat ovat vaikeimpia NP-ongelmia, mutta niillä on seuraava ominaisuus: Jos tehokas algoritmi yhdelle niistä löytyy, se soveltuu myös muiden NP-ongelmien ratkaisuun. (Scott Aaronson 2008)

Shor käytti algoritmissaan tietynlaisia matemaattisia ominaisuuksia, jotka tuottavat konstruktiivista ja destruktiivista interferenssiä, joita kvanttitietokone tarvitsee toiminnassaan. NP-täydelliset ongelmat eivät näyttäisi sisältävän kyseisiä erikoisominaisuuksia. Tähän päivään mennessä onkin löydetty vain muutamia kvanttialgoritmeja, jotka tuottavat ongelmien ratkaisussa nopeutuksen eksponentiaalisesta ajasta polynomiseen aikaan. (Scott Aaronson 2008)

Valitettavasti kysymykseksi jää edelleen: *Onko olemassa tehokasta kvanttialgoritmia, joka ratkaisee NP-täydellisiä ongelmia?*

4 Kaupallinen puoli

Raportissa (National Academies of Sciences, Medicine ym. 2019) mainitun ensimmäisen avainhavainnon mukaan on *erittäin epätodennäköistä, että 2020-luvulla rakennettaisiin kvanttitietokone, joka voisi vaarantaa RSA 2048:aan tai vastaavan diskreettiin logaritmiin pohjautuvan julkisen avaimen kryptojärjestelmän.* Kvanttitietokoneet ovat teknologiana edelleen varhaisessa vaiheessa, vaikka ne ovat olleet Gartnerin ilmaantuvien teknologioiden listalla 11 kertaa vuosien 2000 ja 2017 välisenä aikana. Joka kerta ne ovat olleet listattuina hypekäyrällä varhaisimmassa vaiheessa olevina tekniikoina, joiden kaupallistuminen on yli 10 vuoden päässä.

Raportin (National Academies of Sciences, Medicine ym. 2019) toisen avainhavainnon mukaan *valtioiden rahoitusta tarvitaan, jos lähitulevaisuuden kvanttitietokoneet eivät menesty kaupallisesti, estämään merkittävää vähenemistä kvanttilaskentatutkimuksessa ja -kehityksessä.* NISQ-tietokoneissa on todennäköisesti satoja fyysisiä (ei virhekorjattuja) kubitteja, ja vaikka niille on lupaavia tutkimussuuntauksia, tällä hetkellä ei ole tunnettuja algoritmeja tai sovelluksia, jotka ottaisivat tehokkaasti hyödyn tällaisen luokan laitteesta. On tarpeellista luoda tutkimus- ja kehitysohjelma, jonka tavoitteena on kehittää kaupallisia sovelluksia lähitulevaisuuden kvanttilaskennalle. Ohjelma voisi sisältää seuraavia asioita:

1. Tunnistaa algoritmit, joilla on kohtuullinen ongelmakoko ja rajoitettu porttisyvyys havaittavan kvanttinopeutuksen kera, sovellusalueilla, joissa klassisten tietokoneiden algoritmit eivät todennäköisesti juurikaan parane.
2. Tunnistaa algoritmit, joille klassinen-kvanttimainen-hybriditekniikka, käyttäen kohtuullisenkokoista kvanttialijärjestelmää, tuo merkittävästi nopeutusta.
3. Tunnistaa ongelma-alueet, joissa parhaat klassisten tietokoneiden algoritmit ovat parhaillaan törmäämässä klassisen laskennan luontaisiin skaalarajoihin, ja joille kohtuullinen ongelmakoon kasvattaminen tuo ekonomisesti merkittävää kasvua ratkaisuvaikutuksellaan.

4.1 Mitkä ovat odotetut roolit kvanttilaskennassa vs. klassisessa laskennassa 5 - 10 vuoden sisällä?

4.1.1 Reversiibelit tietokoneet

Ennen yleiskäyttöisiä kvanttitietokoneita markkinoille voi tulla klassisia tietokoneita, jotka ovat kuitenkin reversiibeleitä. Mooren laki ei ole 2010-luvun loppupuolella eikä 2020-luvun alkupuolella enää oikein toiminut entiseen malliinsa, koska nykyisenlaisten klassisten tietokoneiden suorituskyky alkaa tulla vastaan. (*The Reversible Computing FAQ (Frequently Asked Questions)*)

On kuitenkin esitetty ajatus reversiibeleistä tietokoneista, joissa laskenta tapahtuu reversiibeleillä operaatioilla. Tällaiset operaatiot voidaan helposti ja eksaktisti kääntää tai kumota. Frankin (Frank 2006) mukaan *reversiibeli laskenta on 1000–100 000 kertaa niin kustannustehokasta kuin ei-reversiibeli laskenta 2050-luvulla.*

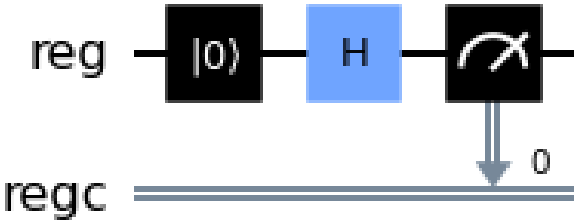
Reversiibeliä laskentaa käsitellään lisää tämän raportin sivulla 47.

4.2 Mitkä ohjelmat / toimialueet kokevat suurimman vaikutuksen?

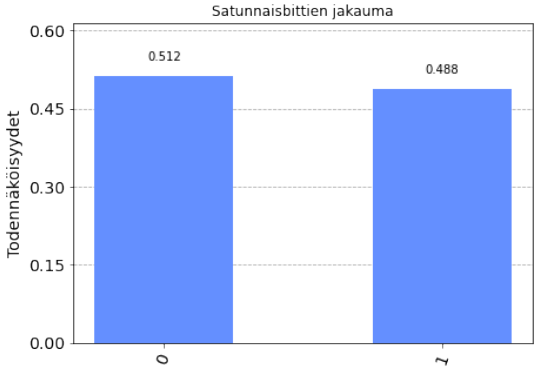
4.2.1 Satunnaislukujen generointi

Klassisilla tietokoneilla luodaan satunnaislukuja, jotka eivät oikeastaan olekaan satunnaislukuja vaan pseudosatunnaislukuja. Pseudosatunnaislukugeneraattorin algoritmin ja parametrit tietämällä voi melko helposti ennustaa, millaisia pseudosatunnaislukuja kone arpoo. Tämä on suuri tietoturvariski.

Kvanttitietokoneilla voidaan luoda oikeasti satunnaisia satunnaislukuja valmistamalla Hadamardin operaattorin avulla kubitista superpositiotila, ja sitten mittaamalla tila. Kuvassa 7 on varsin yksinkertainen kvanttipiiri, joka toteuttaa Hadamardin portin avulla superpositiotilan. Lopuksi mittausoperaatio tuottaa teorian mukaan 50% todennäköisyydellä 0:n ja 50% todennäköisyydellä 1:n. Kuvassa 8 on satunnaisbittijakauma 1000 ajon jälkeen. Ohessa on Python-kielellä ja Qiskitillä tehty toteutus satunnaisbitin tuottamiseen IBM:n kvanttitietokoneella.



Kuvio 7. Kvanttipiiri, jossa kubitti asetetaan Hadamardin portin avulla superpositiotilaan. Mittaus tuottaa 50% todennäköisyydellä nollan ja 50% todennäköisyydellä ykkösen.



Kuvio 8. IBM:n kvanttitietokoneen tuottamassa satunnaisbittijakaumassa päästään 1000 ajon jälkeen lähes tasatuloksiin.

```
## Programming Quantum Computers
##   by Eric Johnston, Nic Harrigan and Mercedes Gimeno-Segovia
##   O'Reilly Media
##
## More samples like this can be found at http://oreilly-qc.github.io
## This sample generates a single random bit.
## Alkuperinen: https://web.archive.org/web/20210119204333/https://raw.
    githubusercontent.com/oreilly-qc/oreilly-qc.github.io/master/samples/
    Qiskit/ch02_01_random_bit.py
## Muokannut: Henri Tapani Heinonen
## https://web.archive.org/web/20201007080747/https://stackoverflow.com/
    questions/34398054/ipython-notebook-cell-multiple-outputs
from qiskit import QuantumCircuit, QuantumRegister, ClassicalRegister,
```

```
    execute, Aer, IBMQ, BasicAer
from qiskit.visualization import plot_histogram
IBMQ.load_account()
import math
from IPython.core.interactiveshell import InteractiveShell
InteractiveShell.ast_node_interactivity = "all"
from IPython.display import display

%matplotlib inline

reg = QuantumRegister(1, name='reg')
reg_c = ClassicalRegister(1, name='regc')
qc = QuantumCircuit(reg, reg_c)

qc.reset(reg)
qc.h(reg)
qc.measure(reg, reg_c)

provider = IBMQ.get_provider('ibm-q')
backend = provider.get_backend('ibmq_16_melbourne')
job = execute(qc, backend, shots=1000)
result = job.result()

counts = result.get_counts(qc)
print('Satunnaisbitit (ihannetapauksessa molempia on 50% ja 50%):',counts
    )

display(qc.draw("mpl"))
fig = plot_histogram(counts, title="Satunnaisbittien jakauma")
fig.axes[0].set_ylabel('Todennäkisyydet')
display(fig)
```

Listing 4.1. Satunnaisbitin generointi IBM:n aidolla kvanttitietokoneella.

4.2.2 Kvanttifysiikan ja kvanttikemian simulointi

Ehkä todennäköisin alue, jossa kvanttitietokonetta käytetään, on kvanttifysiikan ja kvanttikemian ilmiöiden simulointi. Artikkelissa (Lu ym. 2012) keskustellaan kvanttisimulaation ideoista, kvanttisimulaattorien taustoista, kvanttisimulaattorien kategorioista ja sekä teorian että kokeiden kehityksestä. Lisäksi esitellään lyhyesti kvanttikemiaa klassisten tietokoneiden näkökulmasta. Pientä kvanttitietokonetta käyttäen arvioidaan staattista molekyylin ominaisenergiaa ja kemiallisen reaktiodynamiikan simulointia. Odotettavissa on, että tulevaisuudessa kvanttisimuloinnista tulee tehokas työkalu kvanttikemiassa suhteessa klassiseen laskentaan.

ScienceDaily uutisoi läpimurrosta kvanttifysiikan simuloinnissa (New York), jonka mukaan fyysikot ovat löytäneet kvanttialgoritmin, jonka avulla on potentiaalia tutkia monen elektronin kvanttijärjestelmiä. (Rahmani ym. 2020)

4.2.3 Kvantinjälkeinen kryptografia

Kokonaislukujen tekijöihinjakoa voidaan käyttää hyökkäyksissä kryptografisia järjestelmiä vastaan. Klassisilla tietokoneilla nämä hyökkäykset ovat hyvin aikaavieviä, jos kyseiset alkuluvut ovat riittävän suuria. Shorin kvanttialgoritmi (Shor 1994) nopeuttaa tekijöihinjakoa, joten monet nykyiset kryptografiset järjestelmät ovat vaarassa, jos riittävän kyvykäs kvanttitietokone tulee pahantahtoisen tahon haltuun.

Lamportin allekirjoitukset ovat esimerkki “kvantinjälkeisestä kryptografiasta”. Niiden avulla voitaisiin ratkaista esimerkiksi Bitcoinia uhkaavat kvanttitietokonehyökkäykset. Tässä osiossa lähteinä on käytetty Vitalik Buterinin artikkelia Bitcoin Magazine -lehdessä (*Bitcoin Is Not Quantum-Safe, And How We Can Fix It When Needed*), Chris Pacian artikkelia Bitcoin Not Bombs -sivustolla (*Bitcoin vs. The NSA’s Quantum Computer*), Henri Heinosen sivua Lamportin allekirjoituksista (*Bitcoin, kvanttitietokoneet ja kyberturvallisuus*) sekä JP Richardsonin artikkelia (*Generating a Bitcoin Address with JavaScript*), jossa generoidaan bitcoin-osoite JavaScriptillä. Lisäksi aihetta käsitellään lisää tämän raportin liitteessä D sivulla 68.

Bitcoin on P2P-verkossa toimiva hajautettu kryptovaluutta. Se käyttää elliptisen käyrän kryptografiaa secp256k1. Bitcoin-osoitteet ovat kirjaimista ja numeroista koostuva merkkijono,

joka yleensä alkaa joko numerolla 1 tai 3. Bitcoin-osoite on hieman kuin pankkitilin tilinumero: sen perusteella rahaa voidaan lähettää oikealle henkilölle tai taholle. Aivan kuten tilinumeronkin, bitcoin-osoitteen voi turvallisin mielin antaa kenelle vain. Pelkästään sen avulla ei voida bitcoineja käyttää. Jokaista bitcoin-osoitetta kohden on kaksi kryptografista avainta: yksityinen avain ja julkinen avain.

Yksityistä avainta tarvitaan, jotta voi lähettää bitcoineja eteenpäin. Se on usein 256-bittinen satunnaiskokonaisluku.

Julkista avainta tarvitaan, jotta muut Bitcoin-verkon käyttäjät voivat todentaa, että bitcoin-siirto on todellakin peräisin oikealta käyttäjältä. Julkinen avain paljastuu kaikille siinä vaiheessa, kun tekee bitcoin-siirron. Vaikka julkinen avain on luotu yksityisestä avaimesta (elliptisen käyrän kertolaskulla), ei yksityistä avainta voida, klassisilla tietokoneilla, selvittää julkisesta avaimesta, koska ECDSA "toimii yhteen suuntaan".

Bitcoin-osoite muodostetaan tiivistämällä julkinen avain SHA256-algoritmilla, sitten tämä tiiviste vielä tiivistetään uudelleen RIPEMD-160-algoritmilla. Saadun RIPEMD-160-tiivisteen eteen lisätään käytettävän verkon versionumero ja loppuun lisätään tarkistussumma. Saatu lopputulos ajetaan vielä Base58-muotoon, jotta osoitteessa ei olisi epähaluttuja merkkejä kuten 0 (nolla), O (iso o), I (iso i), l (pieni L), + (plus-merkki) ja / (vinoviiva).

ECDSA ei ole erityisen kvanttitietokoneturvallinen, koska Shorin algoritmilla se voidaan hyvin tehokkaasti murtaa. Groverin algoritmilla voidaan puolestaan yrittää murtaa SHA256- ja RIPEMD-160-funktioita, mutta lopputulos ei ole erityisen tehokas: Shorin algoritmi vähentää elliptisen käyrän kryptografian murtamiseen kuluvaa aikaa siten, että $O(2^{k/2})$ onkin jatkossa enää $O(k^3)$. Groverin algoritmi ei vähennä murtamiseen kuluvaa aikaa yhtä dramaattisesti: $O(2^k)$ onkin jatkossa $O(2^{k/2})$. (*Bitcoin Is Not Quantum-Safe, And How We Can Fix It When Needed*)

Jos Bitcoinia yrittää murtaa kvanttitietokoneella, niin heikoin lenkki on ECDSA. Tähän ongelmaan pitää löytää lähiaikoina ratkaisu, jotta bitcoinit ehditään siirtää turvaan, ennen kuin pahantahtoiset tahot saavat käsiinsä kvanttitietokoneita. Pahimmillaan koko bitcoin-talous romahtaa, jos bitcoineja ehditään varastaa suuria määriä tämän tietoturvaongelman johdosta.

Ratkaisu on siirtyminen bitcoin-siirroissa käyttämään Lamportin allekirjoituksia (Lamport 1979). Ohessa on esimerkki, miten Matleena voisi lähettää turvallisesti bitcoineja Henrille käyttäen Lamportin allekirjoituksia.

1. Valmistelu: Matleena luo $160 \times 2 = 320$ satunnaislukua (yksityisavainta) käyttäen kryptografisesti turvallista satunnaislukugeneraattoria; JavaScriptin Math.random() ei ole kryptografisesti turvallinen. Parhaassa tapauksessa Matleena käyttää kvanttitietokonetta oikeiden satunnaislukujen luomiseen kuten kohdassa 4.2.1. Teknisistä rajoituksista johtuen tuotetaan tarvittava määrä satunnaisia bittejä kvanttisimulaattorilla kuten sivulla 34. Yksityisavaimet ovat taulukossa 2. Hän ottaa kaikista yksityisavaimista tiivisteen RIPEMD-160-algoritmin avulla saadakseen 320 julkista avainta. Julkiset avaimet ovat taulukossa 3.
2. Allekirjoitus: Matleena luo bitcoin-transaktion, johon tulevat julkiset avaimet, viesti ja allekirjoitus. Hän kiinnittää edellisessä vaiheessa luodut *320 julkista avainta* transaktioon. Seuraavaksi hän *kirjoittaa viestin* “Matleena lähettää 5 BTC Henrille”. (Todellisuudessa “Matleena lähettää 5 BTC Henrille”-tyyppisen viestin sijasta käytetään tietenkin Bitcoin-protokollan mukaista bitcoin-transaktioviestiä.) Hän tiivistää viestin RIPEMD-160-algoritmin avulla heksadesimaalimuotoon:
   0x1b944748ca1e61295fa504888d28c717eb85cac9.
   Tämä on binäärimuodossa:
   0001101110010100010001110100100011001010000111100110000100
   1010010101111110100101000001001000100010001101001010001100
   0111000101111110101110000101110010101100100 1.
   Saatu binäärijono on pituudeltaan 160 bittiä. Transaktioon tulevaa allekirjoitusta varten Matleena käy läpi tiivistetyn viestin binäärijonon numero numerolta. Jos binääri on 0, hän valitsee parin ensimmäisen yksityisen avaimen (A). Jos binääri on 1, hän valitsee parin toisen yksityisen avaimen (B). Näin on saatu aikaiseksi 160 yksityisavaimesta muodostettu *Lamportin allekirjoitus*. Matleena lähettää bitcoin-transaktion verkkoon, joten kaikki verkon käyttäjät saavat siis 320 julkista avainta (jul1A, jul1B, jul2A, jul2B, jul3A, jul3B, . . . , jul158A, jul158B, jul159A, jul159B, jul160A, jul160B), viestin ("Matleena lähettää 5 BTC Henrille") ja Lamportin allekirjoituksen sisältämät 160 yksityisavainta (yks1A, yks2A, yks3A, . . . , yks158A, yks159A, yks160B).

3. Verifiointi: Louhijat tiivistävät viestin RIPEMD-160-algoritmillä ja muuttavat sen 160-bittiseksi binäärijonoksi. Sitten louhijat tiivistävät Lamportin allekirjoituksen sisältämät 160 yksityisavainta RIPEMD-160-algoritmillä ja saavat sen perusteella 160 Matleenan julkista avainta. Lopuksi louhijat vielä tarkastavat, että nämä julkiset avaimet täsmäävät transaktion mukaan liitettyjen julkisten avainten kanssa binäärijonon mukaisesti. Jos kaikki täsmäävät, louhijat voivat hyväksyä transaktion liitettäväksi uuden lohkon mukana osaksi Bitcoinin lohkoketjua. Henrin Bitcoin-lompakko ilmoittaa lopulta, että lompakon saldoon on tullut viisi uutta bitcoinia!

320 kappaletta satunnaislukuja kvanttisimulaattorista		
Avainparin numero	160-bittinen satunnaisluku/yksityisavain A	160-bittinen satunnaisluku/yksityisavain B
yks1	**0x1c3d57e965ec676d5be 38381beb6acd89ec783cf**	0x926c3878b0f78d51d8f fae867b0fc430f4184013
yks2	**0xab72601e4e0f882d6d0 20bf07f144a8440c1d15e**	0x34643fc63d854464cac 0529e0eb5b838a9cdebd2
yks3	**0x80c67d3b86d5b74d25f 415f2bfec3209770c6a1e**	0x153f2ab7b0722bcc585 263ced55639b5baed493f
...	...	...
yks158	**0xcb12142b853af9164b3 6bdb51212a7c0e63f58c0**	0x3b8f4b6131498a210fa ac4d6ffc0439a2c6f0d38
yks159	**0x422af9cb6fef0627aee 6c883a50aaa84e13c8b8f**	0xfb1825d27af844b56a3 99d11a2e71d1684b0a573
yks160	0x226e109e865f4f5f990 23996fc85c952103fbad6	**0xc24f1933b6c65b5d96d 24c1a834db4f05d40c2a3**

Taulukko 2. 320 kappaletta 160-bittisiä satunnaislukuja toimivat yksityisavaimina. Lihavointi korostaa ne yksityisavaimet, jotka Matleena poimii Lamportin allekirjoitukseen mukaan.

Oheisessa ohjelmalistauksessa tuotetaan Qiskitin avulla satunnaistavu (8 satunnaista bittiä) kvanttisimulaattorilla. 6400 ajon jälkeen on kertynyt tarvittavat 51 200 bittiä. Todellisuudessa ei kannata käyttää kvanttisimulaattoria satunnaislukujen tuottamiseen, koska taustalla toimii klassinen tietokone. Teknisistä rajoituksista johtuen tässä esimerkissä ei käytetty oikeaa

320 kappaletta RIPEMD-160-tiivisteitä satunnaisluvuista		
Avainparin numero	160-bittinen tiiviste / julkinen avain A	160-bittinen tiiviste / julkinen avain B
jul1	0x9b47d2cf573c8b7915c 67a7e8bcec112eb6f941e	0xe98a1ead3ed54431cab bfdad1dc0cb913f6800e6
jul2	0x2733e4b00007ba59b53 0b50e2ac16c7f836e6f81	0x994a0802cdc96f1f562 01cc0d37babfd29f8f3ce
jul3	0xe07798f38a427e6ed26 4c53818018f622e7f12b5	0xaaba2069e2c67f4ae84 5de4664e7e6b31c984165
...	...	...
jul158	0x8560990bce54f38ca0a 1b4f3c011edc41608d094	0xb971750516db4182afa f159c3bfa9ebe66018c73
jul159	0x1075229ccdfc0538f31 e5b08ffa3db673a6b0e8d	0x1983b16a62d2899bca5 7c06ee3e76eff052407e0
jul160	0xd67cc79630cc21039dd d2f409f7feaf79ec194e1	0x8a7e5b5b37b5eb55129 6655698e7b43b163d047b

Taulukko 3. 320 kappaletta 160-bittisiä RIPEMD-160-tiivistettä yksityisavaimista toimivat julkisina avaimina.

kvanttitietokonetta.

```
## Programming Quantum Computers
##   by Eric Johnston, Nic Harrigan and Mercedes Gimeno-Segovia
##   O'Reilly Media
##
## More samples like this can be found at http://oreilly-qc.github.io
## Example 2-2: Random byte
## L hde: https://web.archive.org/web/20210127210016/https://raw.
    githubusercontent.com/oreilly-qc/oreilly-qc.github.io/master/samples/
    Qiskit/ch02_02_random_byte.py
## Muokannut: Henri Tapani Heinonen
from qiskit import QuantumCircuit, QuantumRegister, ClassicalRegister,
    execute, Aer, IBMQ, BasicAer
import math
```

```
import hashlib
h = hashlib.new('ripemd160')

%matplotlib inline

reg = QuantumRegister(8, name='reg')
reg_c = ClassicalRegister(8, name='regc')
qc = QuantumCircuit(reg, reg_c)

qc.reset(reg)
qc.h(reg)
qc.measure(reg, reg_c)

backend = BasicAer.get_backend('qasm_simulator')
job = execute(qc, backend, shots=6400, memory=True)
result = job.result()

counts = result.get_counts(qc)
memory = result.get_memory(qc)
satunnaisluku = ''.join(memory)
len(satunnaisluku) # 51200 bitti

# http://web.archive.org/web/20210127210939/https://www.kite.com/python/answers/how-to-convert-a-binary-string-to-a-hexadecimal-string-in-python
# https://web.archive.org/web/20210127212003/https://stackoverflow.com/questions/9475241/split-string-every-nth-character
yksityisavaimet = [hex(int(satunnaisluku[i:i+160], 2)) for i in range(0, len(satunnaisluku), 160)]
# https://web.archive.org/web/20210127212141/https://stackoverflow.com/questions/42601709/hashlib-cant-find-ripemd160

julkiset_avaimet = []
for i in yksityisavaimet:
    h = hashlib.new('ripemd160')
    h.update(i.encode('utf-8'))
    julkiset_avaimet.append("0x"+h.hexdigest())
```

```
43
44 print("yksityisavain 1a = ", yksityisavaimet[0], ", julkinen avain 1a = "
    , julkiset_avaimet[0])
45 print("yksityisavain 1b = ", yksityisavaimet[1], ", julkinen avain 1b = "
    , julkiset_avaimet[1])
46 print("yksityisavain 2a = ", yksityisavaimet[2], ", julkinen avain 2a = "
    , julkiset_avaimet[2])
47 print("yksityisavain 2b = ", yksityisavaimet[3], ", julkinen avain 2b = "
    , julkiset_avaimet[3])
48 print("yksityisavain 3a = ", yksityisavaimet[4], ", julkinen avain 3a = "
    , julkiset_avaimet[4])
49 print("yksityisavain 3b = ", yksityisavaimet[5], ", julkinen avain 3b = "
    , julkiset_avaimet[5])
50 print("...")
51 print("yksityisavain 158a = ", yksityisavaimet[-6], ", julkinen avain 158
    a = ", julkiset_avaimet[-6])
52 print("yksityisavain 158b = ", yksityisavaimet[-5], ", julkinen avain 158
    b = ", julkiset_avaimet[-5])
53 print("yksityisavain 159a = ", yksityisavaimet[-4], ", julkinen avain 159
    a = ", julkiset_avaimet[-4])
54 print("yksityisavain 159b = ", yksityisavaimet[-3], ", julkinen avain 159
    b = ", julkiset_avaimet[-3])
55 print("yksityisavain 160a = ", yksityisavaimet[-2], ", julkinen avain 160
    a = ", julkiset_avaimet[-2])
56 print("yksityisavain 160b = ", yksityisavaimet[-1], ", julkinen avain 160
    b = ", julkiset_avaimet[-1])
```

Listing 4.2. Satunnaistavun tuottaminen kvanttisimulaattorilla. 6400 ajon jälkeen on kertynyt tarvittavat 51 200 bittiä.

Lamportin allekirjoitus on toimiva keino kvanttitietokonehyökkäyksiä vastaan, sillä Groverin algoritmilla tarvitaan 2^{80} askelta väärennetyn transaktion luomiseen. (*Bitcoin Is Not Quantum-Safe, And How We Can Fix It When Needed*)

Lamportin allekirjoitukset ovat vain yksi esimerkki kvantinjälkeisestä kryptografiasta. Lisää tietoa kvantinjälkeisestä kryptografiasta löytyy kirjallisuudesta (Bernstein, Buchmann ja Dahmen; Ding ja Steinwandt 2019; Ding ja Tillich 2020).

4.2.4 Ohjattu ja ohjaamaton oppiminen sekä kvanttivahvistusoppiminen kvanttitietokoneilla

Kvanttikoneoppiminen on uusi tieteenala; siinä yhdistävät koneoppiminen ja kvanttilaskenta. Ohjattua oppimista kvanttitietokoneilla (engl. "supervised learning with quantum computers") käsitellään esimerkiksi kirjassa (Schuld ja Petruccione 2018). On olemassa useita määritelmiä kvanttikoneoppimiselle.

Yksi määritelmistä jakaa kvanttikoneoppimisen neljään kategoriaan: CC, CQ, QC ja QQ. Kirjainyhdistelmän ensimmäinen kirjain tarkoittaa, onko dataa generoiva järjestelmä klassinen (C) vai kvanttimainen (Q). Kirjainyhdistelmän toinen kirjain tarkoittaa, onko dataa prosessoiva järjestelmä klassinen vai kvanttimainen.

CC on tietysti perinteinen koneoppiminen, jossa klassista dataa prosessoidaan klassisesti. Koneoppimisen metodit voivat tässäkin tapauksessa olla peräisin kvantti-informaatiotutkimuksesta.

QC:ssä tutkitaan, miten koneoppiminen voisi auttaa kvanttilaskennassa. Esimerkiksi, jos halutaan tietää kvanttitietokoneen sisäinen tila mahdollisimman vähin mittauksin, voidaan käyttää koneoppimista analysoimaan mittausdataa.

CQ:ssä käytetään kvanttilaskentaa prosessoimaan klassisia datajoukkoja. Voidaan esimerkiksi syöttää kvanttitietokoneelle klassista dataa kuten tekstiä, kuvia tai makroekonomian muuttujien aikasarjoja. Tutkimuksen tavoitteena on siis luoda kvanttialgoritmeja tiedonlouhinnalle.

Kirjassa (Schuld ja Petruccione 2018) keskitytään ohjattuun oppimiseen CQ-kategoriassa, mutta on olemassa myös kvanttivahvistusoppimista (engl. "quantum reinforcement learning") sekä ohjaamatonta oppimista kvanttitietokoneilla (engl. "unsupervised learning with quantum computers"). Ohjatussa oppimisessa on käsillä datajoukko, jossa on ratkaisuja ongelmaan, ja näitä ratkaisuja käytetään ohjauksessa uutta ongelmaa ratkaistaessa.

QQ:ssä kvanttidataa käsitellään kvanttitietokoneella. Tästä kategoriasta ei vielä ole paljon tutkimustuloksia. Kvanttitietokone voisi simuloida kvanttijärjestelmän dynamiikkaa kuten kohdan 4.2.2 esimerkeissä. Tämän kvanttijärjestelmän tila syötettäisiin lopulta kvanttikoneoppimisalgoritmille.

5 Lyhyt kuvaus kvanttitietokonelaitteiston kehityksen nykytilasta ja tulevaisuudesta

Tämän luvun taulukkotiedot pohjautuvat enimmäkseen Wikipedian artikkeliin QPU-järjestelmistä. (*List of quantum processors - Wikipedia*)

Raportissa (National Academies of Sciences, Medicine ym. 2019) jaetaan nykyiset kvanttitietokoneet kolmeen kategoriaan: “analogiset kvanttitietokoneet”, “digitaaliset NISQ-tietokoneet” ja “täysin virheenkorjatut kvanttitietokoneet”. Analogiset kvanttitietokoneet manipuloivat kubittien välisiä vuorovaikutuksia suoraan jakamatta näitä toimintoja primitiivisiin porttioperaatioihin. Esimerkkejä analogisista kvanttitietokoneista ovat anniilauskvanttisuorittimet, adiabaattiset kvanttitietokoneet ja suorat kvanttisimulaattorit. Digitaaliset NISQ-tietokoneet operoivat ajamalla halutun algoritmin käyttämällä primitiivisiä porttioperaatioita fyysisiin kubitteihin. Molemmissa konetyypeissä on kohinaa. Täysin virheenkorjatut kvanttitietokoneet ovat porttipohjaisia kvanttitietokoneita, jotka ovat robustimpia siksi, että niihin sovelletaan kvanttivirheenkorjausta (QEC), joka mahdollistaa sen, että kohinaiset fyysiset kubitit emuloivat stabiileja loogisia kubitteja, jolloin tietokone käyttäytyy mille tahansa laskennalle luotettavasti.

Forbesin artikkelin (Moor Insights and Strategy ja Smith-Goodson, Paul) mukaan Xanadu ja PsiQuantum käyttävät integroitua fotoniikkaa; Google, IBM ja Rigetti käyttävät suprajohtavia materiaaleja ja Honeywell ja IonQ käyttävät ioniloukkuja kubittien luomiseen.

5.1 IBM

IBM:llä on muutamia kvanttitietokonejärjestelmiä (*IBM Quantum systems overview*).

Korkealuokkaiset piiripohjaiset kvanttisuorittimet		
Nimi	Kubittien määrä	Kvanttitilavuus
Manhattan	65	32
Rochester	53	8
Cambridge	28	8
Montreal	27	32
Paris	27	32
Toronto	27	32
Athens	5	32
Bogota	5	32
Rome	5	32

Avoimet piiripohjaiset kvanttisuorittimet		
Nimi	Kubittien määrä	Kvanttitilavuus
Melbourne	15	-
Santiago	5	32
Ourense	5	8
Vigo	5	16
Valencia	5	16
Yorktown (QX2)	5	8
Armonk	1	-

Kvanttitilavuus on metodi laitteen suorituskyvyn verifiointiin ja metriikka kvanttilaitteen laskentatehon määrälliseen ilmaisuun. (*Quantum Volume – Qiskit 0.23.3 documentation; Quantum Volume: The Power of Quantum Computers*)

IBM:n Quantum Experience käyttää kubittina suprajohtavaa transmonikubittia, joka on tehty suprajohtavasta materiaalista kuten niobiumista ja alumiinista, jotka on kuvioitu piialustaan. Ne eivät ole luonnostaan kubitteja, mutta monista energiatasoista kaksi eristämällä saadaan muodostettua likimääräinen kubitti. (*The qubit - Docs and Resources - IBM Quantum Experience*)

5.2 Google

Googlen tutkijat ilmoittivat vuonna 2019 saavuttaneensa kvanttiherruuden (Arute ym. 2019). Oheisen listauksen tiedot ovat Googlen blogista (*Quantum Supremacy Using a Programmable Superconducting Processor; A Preview of Bristlecone, Google's New Quantum Processor*).

Piiripohjaiset kvanttisuorittimet		
Nimi	Kubittien määrä	Julkaisuvuosi
Bristlecone	72	2018
Sycamore	54	2019

5.3 Rigetti Computing

Rigetti Computing on Kaliforniassa sijaitseva yritys, joka kehittää mikropiirejä kvanttitietokoneita varten. Yrityksellä on myös pilvipalvelu nimeltään Forest, joka mahdollistaa kvanttialgoritmien kirjoittamisen. Oheisen listauksen tiedot ovat Rigettin kotisivuilta (*Scalable quantum systems built from the chip up to power practical applications*).

Piiripohjaiset kvanttisuorittimet		
Nimi	Kubittien määrä	Julkaisuvuosi
Agave	8	2018
Acorn	19	2017
Aspen-1	16	2018
Aspen-4	13	2019
Aspen-7	28	2019

5.4 D-Wave Systems, Inc.

D-Wave Systems on kanadalainen yritys, joka oli ensimmäisenä maailmassa myymässä tietokoneita, jotka hyödynsivät kvantti-ilmiöitä. Vuonna 2015 esiteltiin D-Wave 2X -kvanttitietokone, jossa oli yli 1000 kubittia. Kyseessä on anniilauskvanttitietokone. Oheisen listauksen tiedot ovat Wired-lehden artikkelista (*D-Wave Defies World of Critics With 'First Quantum Cloud'*) ja Next Big Future -sivustolta (*Dwave Systems shows off quantum chip with 2048 physical*

qubits).

Anniilauskvanttisuorittimet		
Nimi	Kubittien määrä	Julkaisuvuosi
D-Wave One (Ranier)	128	2011
D-Wave Two	512	2013
D-Wave 2X	1152	2015
D-Wave 2000Q	2048	2017

5.5 Kiinan tiede- ja teknologiayliopisto (USTC)

Kiinan tiede- ja teknologiayliopiston Jiuzhang on fotonipohjainen, 76-kubittinen, kvanttitietokone, joka saavutti vuonna 2020 kaikkien aikojen toisen kvanttiherruuden - Googlen suprajohtavia materiaaleja hyödyntävän kvanttitietokoneen vuoden 2019 kvanttiherruuden jälkeen. Jiuzhang-kvanttitietokone suoritti 200 sekunnin aikana laskennan, joka olisi vienyt puoli miljardia vuotta maailman tehokkaimmalta, klassiselta, supertietokoneelta. (Zhong ym. 2020; Conover).

Piiripohjaiset kvanttisuorittimet		
Nimi	Kubittien määrä	Julkaisuvuosi
Jiuzhang	76	2020

5.6 Xanadu

Myös Xanadu käyttää fotonipohjaista kvanttitietokonetekniikkaa. Heidän mukaansa kyseinen tekniikka on skaalautuvaa (käyttää helposti valmistettavaa piipohjaista tekniikkaa), robustia (kubitit ovat virheensietäviä) ja käytännöllistä (toimii ensisijaisesti huoneenlämmössä). (*Hardware - Xanadu*)

Piiripohjaiset kvanttisuorittimet		
Nimi	Kubittien määrä	Julkaisuvuosi
X8	8	2020

5.7 Honeywell

Honeywellin mukaan se on julkaissut vuoden 2020 aikana kvanttitietokoneen, jonka kvanttitilavuus on ainakin 64. Kvanttitilavuus riippuu ainakin kubittien lukumäärästä (engl. "number of qubits"), virhesuhteesta (engl. "error rate") ja kubittien liitettävyydestä (engl. "connectivity of qubits"). (*Quantum Volume: The Power of Quantum Computers*)

5.8 PsiQuantum

PsiQuantum on kalifornialainen yritys, joka pyrkii fotoniikan avulla rakentamaan ensimmäisen käyttökelpoisen kvanttitietokoneen, jossa on jopa miljoona kubittia. (Atomico)

5.9 IonQ

IonQ on marylandilainen vuonna 2015 perustettu yritys, joka pyrkii ioniloukkujen avulla rakentamaan kvanttitietokoneen, jossa on 32 kubittia ja kvanttitilavuutta ainakin 4 000 000. (Chapman)

5.10 VTT ja IQM

VTT on Suomen valtion omistama yhtiö ja IQM on suomalainen startup-yritys. Lehdistötiedotteen (Vasara, Antti ja Majumdar, Himadri) mukaan ne ovat allekirjoittaneet sopimuksen innovaatiokumppanuudesta Suomen ensimmäisen kvanttitietokoneen rakentamiseksi. Suomen hallitus rahoittaa projektia 20,7 miljoonalla eurolla. Ensimmäisessä vaiheessa on tavoitteena toimiva viiden kubitin kvanttitietokone. Kokonaistavoitteena on 50 kubitin kvanttitietokone vuoteen 2024 mennessä.

5.11 Kvanttilaitteistojen arvioidut kapasiteetit ja rajoitukset 5 - 10 vuoden sisällä

Dyakonovin (Dyakonov 2020) mukaan universaalia, käyttökelpoista, kvanttitietokonetta ei todennäköisesti koskaan saada rakennettua. Hän perustelee väitteen sillä, että jo 1000 kont-

rolloitavan parametrin klassisen analogisen tietokoneen rakentaminen on ylivoimainen tehtävä. Tähän tarvitaan $2 \cdot 1000$ jatkuvaa parametria. 1000 kubitin kvanttitietokone olisi vielä haastavampi operaatio, koska sellaisen laitteen tilan kuvantamiseen tarvitaan 2^{1000} jatkuvaa parametria.

Aaronsonin (Scott Aaronson 2008) mukaan yleinen harhakäsitys on, että kvanttitietokoneet voisivat ratkaista NP-täydellisiä ongelmia kokeilemalla jokaista mahdollista ratkaisua ongelmaan samanaikaisesti. Tällainen taianomainen tietokone voisi löytää ratkaisun vaikkapa Goldbachin konjektuuriin tai Riemannin hypoteesiin etsimällä kaikki mahdolliset todistukset ja vastatodistukset, joissa on korkeintaan, esimerkiksi, miljardi symbolia. Toisaalta, yhtälailla ei ole hyvä suhtautua kvanttilaskentaan kuin tieteiskuvitelmaan. On tärkeää yrittää löytää kvanttitietokoneiden rajat ja mahdollisuudet.

Uuden tutkimuksen (Vepsäläinen ym. 2020; *Future quantum computers may have to be built UNDERGROUND*) mukaan ionisoiva säteily, joka on peräisin kosmisesta säteilystä ja luonnon omasta radioaktiivisuudesta, aiheuttaa suprajohtavissa kubiteissa häiriöitä rajoittaen koherenssiaikoja millisekunteihin. Yhtenä ratkaisuna esitetään kvanttitietokoneiden rakentamista syvälle maan alle. Toinen ratkaisu voisi olla kubittien rakentaminen vähemmän ionisoivalle säteilylle herkästä materiaalista: ainakin USTC, Xanadu ja PsiQuantum käyttävät fotoniikkaan pohjautuvia menetelmiä kubittien luomiseen.

6 Muun aiheeseen liittyvän tutkimuksen hahmotelma

Kvanttilaskennan ja perinteisen binäärimuotoisen klassisen laskennan lisäksi on tärkeää ymmärtää muitakin laskennan muotoja: Digitaalinen tietokone voi hyödyntää esimerkiksi kolmiarvoista logiikkaa (ternääri/trinääri). Ei-reversiibeleistä porteista (JA, TAI, jne.) voidaan tehdä reversiibelit versiot (R-JA, R-TAI, jne.), jolloin laskenta olisi reversiibeliä ja tehonkulutus nykyistä pienempää. Tietokoneet voivat olla myös analogisia. Taulukkoon 4 on koottu laskennan historian merkkipaaluja.

6.1 Analoginen laskenta

Analoginen tietokone on fyysinen järjestelmä, joka on konfiguroitu siten, että sitä hallinnoivat yhtälöt ovat identtisiä niiden yhtälöiden kanssa, jotka halutaan ratkaista. Alkutila asetetaan samanlaiseksi kuin tarkasteltavalla ongelmalla. Sitten analogisen tietokoneen muuttujien annetaan kehittyä ajan suhteen. Yksinkertainen esimerkki mekaanisesta analogisesta tietokoneesta olisi vesiletku ja ämpäri. Tarkasteltava matemaattinen ongelma olisi integraalifunktio. Letkun veden virtaamisnopeus asetetaan samaksi kuin funktio, joka halutaan integroida. Vesi ohjataan ämpäriin. Matemaattisen ongelman ratkaisu on lopulta ämpärissä olevan veden määrä. (Tsividis 2018)

Vanhin tunnettu tietokone on Antikytheran kone eli eräänlainen analoginen mekaaninen tietokone, joka laskee tähtitieteellisiä laskuja. Se on luultavasti valmistettu 100 vuotta ennen ajanlaskun alkua, ja se löydettiin Antikytheran laivan hylystä vuonna 1901. (Connelly 2008; Caravelli ja Carbajal 2018)

Esimerkkinä modernimman aikakauden sähkömekaanisesta analogisesta tietokoneesta on noin vuoteen 1943 ajoittuva TDC Mark III, jota käytettiin tietyissä sukellusveneiden operaatioissa. (Connelly 2008)

Apollo-lentojen, ja niitä edeltäneiden avaruuslentojen, onnistumisten kannalta avainasemassa olivat analogiset tietokoneet ja hybriditietokoneet (analogis-digitaaliset), joita NASA käytti simuloinneissa ja jopa lennonohjauksessa. Analogiset ja analogis-digitaaliset tietokoneet

Muutamia laskennan historian merkkipaaluja	
Vuosi	Teknologia tai keksintö ja sen selitys
∼100 eaa.	Antikytheran kone on ensimmäinen tunnettu analoginen ja mekaaninen tietokone
∼1943	TDC Mark III on esimerkki analogisesta sähkömekaanisesta tietokoneesta
1945	ENIAC oli ensimmäinen ohjelmoitava, sähköinen, yleiskäyttöinen digitaalinen tietokone
1958	Setun oli ensimmäinen moderni, sähköinen, ternääritietokone
1961	Rolf Landauer havaitsi, että informaation hävittäminen haihduttaa energiaa
1969	Analogiset ja analogis-digitaaliset tietokoneet osaltaan mahdollistivat ihmisen matkan Kuuhun
1973	Charles Bennet havaitsi, että klassinen laskenta on mahdollista muuttaa reversiibeliin muotoon
1980	Tommaso Toffoli keksi universaalin reversiibelin logiikkaportin
1980	Paul Benioff esitteli kvanttimekaanisen mallin Turingin koneesta
1993	Cynthia Dwork ja Moni Naor esittelivät työtodistuksen (engl. “Proof-of-Work”) konseptin. (Dwork ja Naor 1993)
1994	Peter Shor esitteli kvanttialgoritmit diskreeteille logaritmeille ja kokonaislukujen tekijöihinjakamiseen
1999	SETI@home-näytönsäästäjä teki hajautetun luppolaskennan tunnetuksi suurelle yleisölle
2009	Bitcoin aloitti, ja teki työtodistuksen (engl. “Proof-of-Work”) tutuksi suurelle yleisölle vuosia myöhemmin
2012	Kvanttitietokone jakoi Shorin algoritmilla alkulukutekijöihinsä luvun 21 (Martin-Lopez ym. 2012)
2019	IBM:n kvanttitietokone jakoi Shorin algoritmilla alkulukutekijöihinsä luvun 35 (Amico, Saleem ja Kumph 2019)
2019	Googlen tutkijat ilmoittivat saavuttaneensa kvanttiherruuden (Arute ym. 2019)

Taulukko 4. Muutamia laskennan historian merkkipaaluja.

hävisivät, koska niitä oli vaikea suunnitella, rakentaa, käyttää ja ylläpitää 1970-luvun tekniikoilla. Nykytekniikoilla analogiset tietokoneet voivat olla taas käyttökelpoisia, ja niitä tutkitaankin nyt koneoppimisen, koneälyn ja biomimeettisten piirien kontekstissa. (Tsividis 2018)

Analoginen laskenta haastoi vahvan Churchin–Turingin teesin, jonka mukaan mitä tahansa algoritmista prosessia voidaan simuloida tehokkaasti Turingin koneella. Aikoinaan nimittäin ajateltiin, että tietyntyyppisiä analogisia tietokoneita voitaisiin käyttää ratkaisemaan tehokkaasti ongelmia, joilla ei arveltu olevan tehokasta ratkaisua Turingin koneella. Näiden siis arveltiin rikkovan Churchin–Turingin teesin vahvaa muotoa. Lopulta kuitenkin havaittiin, että kaikkien tunnettujen analogisten tietokoneiden tehokkuus häviää kohinan takia realistisissa tilanteissa. Myös kvanttilaskennassa oli alun perin ongelmana laskennan tehokkuuden häviäminen kohinan takia. Lopulta keksittiin kvanttivirheenkorjauskoodit ja vikasietoinen kvanttilaskenta. Ainakin periaatteessa kvanttilaskenta sietää äärellisen määrän kohinaa silti säilyttäen laskennallisen etumatkan. (Nielsen ja Chuang 2010)

Memristoreita käsittelevän tutkimuksen (Caravelli ja Carbajal 2018) mukaan tulevaisuuden tietokoneet ovat todennäköisesti yhdistelmä klassisesti analogisia ja klassisesti digitaalisia tai kvanttimaisesti analogisia ja kvanttimaisesti digitaalisia laskentasiruja.

6.2 Ternäärinen laskenta

Maailmalla on ollut käytössä muutamia ternääritietokoneita, joiden laskenta perustuu ternäärilogiikkaan eli kolmiarvoiseen logiikkaan. Ternäärisessä laskennassa on kolme diskreettiä tilaa: trittiä. Ne voidaan määritellä lukuisilla eri tavoilla

- balansoimaton/tasapainottamaton ternääri/trinääri: $\{0,1,2\}$
- fraktionaalinen balansoimaton/tasapainottamaton ternääri/trinääri: $\{0,1/2,2\}$
- balansoitu/tasapainotettu ternääri/trinääri: $\{-1,0,1\}$
- tuntemattoman tilan logiikka: $\{E,?,T\}$
- ternäärikoodattu/trinäärikoodattu binääri: $\{T,E,T\}$

Edellisessä listassa “T” tarkoittaa tosi-arvoa, “E” tarkoittaa epätosi-arvoa ja “?” tarkoittaa

tuntematonta (samanaikaisesti T ja E). (Connelly 2008)

Kolmiarvoinen logiikka voisi (Balla ja Antoniou 1984; Smith 1981; Porat 1969; Connelly 2008)

- vähentää logiikkafunktioiden toteuttamiseen tarvittavien yhteyksien määrää
- pienentää sirujen pinta-alaa
- sallia siirrettävän enemmän informaatiota tietyn linjan yli
- kuluttaa vähemmän muistia tietylle datapituudelle
- sallia sarja-operaatioille korkeammat nopeudet

Tutkimus (Cambou ym. 2018) selvittää, voisiko ternäärinen laskenta parantaa tietoturvallisuutta. Kvanttikuvankäsittelyä ternääristä kvanttilogiikkaa eli kutrittejä käyttämällä tutkitaan artikkelissa (Caraiman ja Manta 2013).

6.3 Reversiibeli laskenta

2020-luvulla tietokoneiden perusmuistikomponentit ovat atomin kokoluokkaa, jos Mooren laki pitää paikkaansa. Nanoskaalassa nykyinen klassinen laskentateoria lakkaa toimimasta, ja tilalle ilmaantuu kvanttilaskenta, joka vaatii tietokonetieteiden ja informaatioteorian uudelleenkeksimistä siten, että ne ovat yhteensopivia kvanttifysiikan kanssa (Al-Rabadi 2004). Reversiibeleiden tietokoneiden suorittaman laskennan virrankulutus olisi minimalistista nykyisiin ei-reversiibeleihin tietokoneisiin verrattuna. Reversiibeli laskenta on myös voimakkaasti yhteyksissä kvanttilaskentaan.

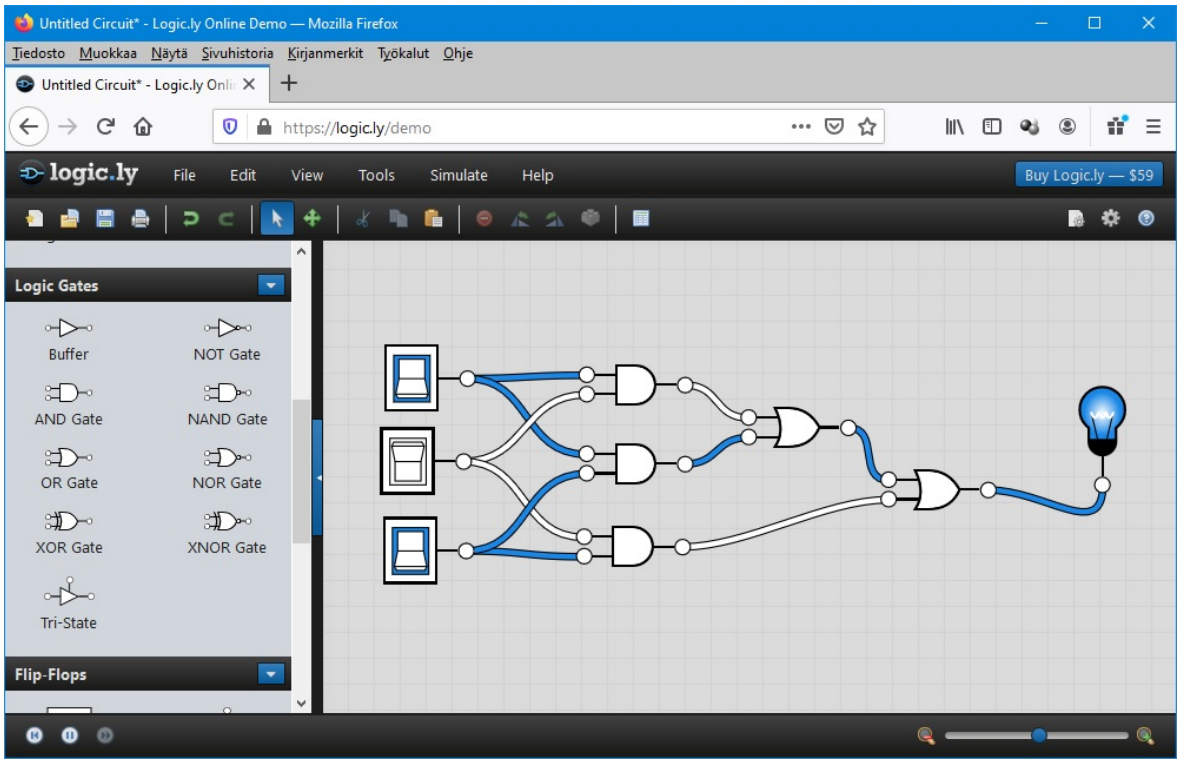
Vuonna 1961 Rolf Landauer havaitsi (Landauer 1961; Kerntopf; Nielsen ja Chuang 2010), että loogisesti ei-reversiibeli portti haihduttaa energiaa ympäristöönsä. Tämä johtuu siitä, että yhden bitin informaation menettäminen haihduttaa energiaa yhtälön

$$E = k_{\mathrm{B}} T \ln(2) \tag{6.1}$$

mukaisesti. Yhtälössä (6.1) k_{B} on Boltzmannin vakio, T on tietokoneen ympäristön lämpötila kelvineissä ja $\ln(2)$ on luonnollinen logaritmi luvusta 2.

Vuonna 1973 IBM:n Charles Bennett havaitsi (Bennett 1973), että mikä tahansa klassinen

laskenta voidaan muuttaa reversiibeliin muotoon. Mitä on reversiibeli laskenta? Oheinen kuvaus on IBM:n kvanttilaskennan oppaasta (*Docs and Resources - IBM Quantum Experience - Shor's algorithm*). Oletetaan, että $f(x)$ on jotain klassista laskentaa, jonka syötteenä on n-bittinen teksti x ja tulosteena m-bittinen teksti $f(x)$. Funktio $f(x)$ voidaan laskea ilman välitulosten pyyhkimistä, jos käytämme ylimääräistä muistia. Datan pyyhkiminen kuluttaa energiaa eli tuottaa hukkalämpöä, joten sitä pyritään välttelemään! Luodaan algoritmi laskennalle $f(x)$ ja muutetaan se sitten alkeellisten logiikkaporttien (JA, TAI, jne.) sarjaksi.



Kuvio 9. Majoriteettipiiri, joka palauttaa tosi-arvon, jos ja vain jos enemmän kuin 50% sen syötteistä on arvoa "tosi".

Majoriteettifunktiolla on n syötettä ja 1 tuloste. Tulos on epätotta, jos $n/2$ tai enemmän syötteistä on epätotta; muutoin tulos on totta. Majoriteettiportti siis palauttaa arvon "tosi", jos ja vain jos enemmän kuin 50% syötteistä on arvoa "tosi". Kuvassa 9 on 3-bittinen majoriteettifunktio toteutettuna kolmella JA-portilla ja kahdella TAI-portilla. Kuvassa valokatkaisimet x_1 ja x_3 ovat päällä, joten enemmän kuin 50% syötteistä on arvoa "tosi". Siispä piiri palauttaa arvon "tosi" ja lamppu syttyy!

Tehdään 3-bittisestä majoriteettipiiristä reversiibeli! Otaksutaan, että jokaisella portilla on kaksi syötejohdinta ja yksi tulostejohdin. Olkoon L porttien kokonaismäärä. Laajennetaan n-bittistä muistia, joka tallettaa syötearvoa x, lisäämällä L bittiä alustettuina nollilla. Nämä ekstrabitit toimivat eräänlaisena "suttupaperina" väliaikatulosten tallentamista varten. i:nnen portin tuloste kirjoitetaan suttumuistin i:nteen bittiin, ja säilytetään syötebittien arvot. Lopuksi funktion $f(x)$ lopputulos sisältyy johonkin suttumuistin kohtaan. Muut kohdat suttumuistista puolestaan sisältävät "roskaa" eli väliaikaista dataa, jota kuvaa bittimerkkijono $g(x)$.

```
x(1) x(2) x(3) suttumuisti..........
|1|  |0|  1    0    0    0    0    0
                                          JA(1,0)=0
|1|  0   |1|  *0*   0    0    0    0
                                          JA(1,1)=1
 1  |0|  |1|   0   *1*   0    0    0
                                          JA(0,1)=0
 1   0    1   |0|  |1|  *0*   0    0
                                          TAI(0,1)=1
 1   0    1    0    1   |0| |*1*|  0
                                          TAI(0,1)=1
 1   0    1    0    1    0    1   *1*
```

Listing 6.1. Majoriteettipiirin tekeminen reversiibeliksi käyttäen suttumuistia.

Suttumuistilistauksen viimeinen rivi sisältää bitit: 10101011, josta 101 on alkuperäiset syötebitit, 0101 on roskaa $g(x)$ ja listauksen viimeinen bitti 1 on lopputulos $f(x)$.

Piiri on nyt kokonaisuudessaan reversiibeli, mutta yksittäiset portit ovat edelleen ei-reversiibeleitä (irreversiibeleitä). Vuonna 1980 Tommaso Toffoli keksi universaalin reversiibelin logiikkaportin (Toffoli 1980), joka tarkoittaa sitä, että mikä tahansa reversiibeli piiri voidaan rakentaa Toffolin porteista. Se tunnetaan myös "kontrolloituna-kontrolloituna EI-porttina". Siinä on 3-bittinen syöte ja 3-bittinen tuloste. Jos ensimmäiset kaksi bittiä ovat molemmat 1, kolmas bitti käännetään ikään kuin se menisi EI-portin läpi. Muissa tapauksissa, kolmannelle bitille ei tehdä mitään.

6.4 Uudet elektroniikan peruskomponentit: memristori, memkondensaattori, memkäämi

Elektroniikassa on tyypillisesti puhuttu kolmesta peruskomponentista: vastuksesta (resistorista), kondensaattorista ja käämistä (kelasta). Nämä ovat jo satoja vuosia vanhoja keksintöjä. 1970-luvulla ennustettiin elektroniikan neljännen peruskomponentin olemassaolo. Sen nimeksi annettiin memristori (muistivastus). Sittemmin on vielä luotu teoriaa kahdelle muullekin uudelle muistikomponentille, joita ovat memkondensaattori (muistikondensaattori) ja memkäämi (memkela, muistikäämi, muistikela) (Radwan ja Fouda 2015; Chua 1971; Di Ventra, Pershin ja Chua 2009; Ventra ja Pershin 2013).

Memristori on laite, joka "käyttäytyy jokseenkin kuin epälineaarinen vastus, jolla on muisti". Sen resistanssi riippuu siihen syötetyn jännitteen tai virran historiasta. Jos syöte poistetaan, ja komponentti jätetään rauhaan, sen resistanssi on seuraavalla käyttökerralla samankaltainen kuin silloin, kun se jätettiin rauhaan. Memristori käyttäytyy siis kuin haihtumaton muisti. Se reagoi myös virran suuntaan, eli memristorilla on polariteetti. (Caravelli ja Carbajal 2018) Memristorista arvellaan olevan hyötyä esimerkiksi vain vähän energiaa kuluttavissa muisteissa ja neuromorfisessa laskennassa. (Peotta ja Di Ventra 2014)

Lienee siis hyödyllistä tutkia, miten muistikomponentit (memristori, memkondensaattori ja memkäämi) voisivat edistää kvanttilaskentaa.

6.5 Klassinen, digitaalinen, binäärinen, ei-reversiibeli laskenta

"Tavallinen" laskenta, joka on klassista, digitaalista, binääristä ja ei-reversiibeliä, on edelleen tärkeää eikä kaikkea potentiaalia ole vielä hyödynnetty. 1990-luvun loppupuolella SETI@home-projekti esitteli näytönsäästäjän, joka Internet-yhteyden yli haki palvelimelta käyttäjän koneelle dataa laskettavaksi. SETI@home etsi Arecibon radioteleskooppidatan seasta merkkejä vieraiden sivilisaatioiden lähettämistä radiosignaaleista. SETI@home lienee maailman tunnetuin luppolaskentaprojekti, joka hyödyntää tavallisen kotitietokoneen luppoajan tieteelliseen laskentaan. Näytönsäästäjästä syntyi BOINC-alusta, jolle on tehty jo kymmeniä luppolaskentaprojekteja: ilmastonmuutosta tutkii Climateprediction.net, IBM:llä on World Community Grid, jossa on ollut parisenkymmentä erilaista hajautetun laskennan projektia puh-

taan energian projektista AIDS-tutkimukseen ja syöpälääkeainelaskentaan.

Kuinka tehokas supertietokone saataisiin aikaan, jos miljardien älypuhelimien ja tietokoneiden luppoaika kanavoitaisiin esimerkiksi BOINC:in kaltaisen alustan kautta tieteelliseen laskentaan? Entäs, jos bitcoin-louhinnassa tehtäisiin muutos työtodistuskonsensusalgoritmiin (engl. “Proof-of-Work consensus algorithm”) siten, että SHA256-laskennan sijasta ratkottaisiin oikeita tieteellisiä ongelmia?

7 Yhteenveto

Tähän yhteenvetoon on lyhyesti koottu raportin kirjoittamisen aikana saatuja havaintoja ja niiden pohjalta syntyneitä johtopäätöksiä ja suosituksia.

- Kvanttitietokoneet, tällä hetkellä, muistuttavat hieman Bitcoin-louhinnassa käytettyjä ASIC-piirejä. Bitcoin-ASIC tekee pelkkää SHA256-laskentaa, mutta se on tosi hyvä siinä. Kvanttitietokonetta voi käyttää yleiskäyttöisenä CPU:na, mutta se ei ole useimmissa tapauksissa klassista tietokonetta nopeampi. Tästä syystä ei aluksi kannata käyttää kohtuuttoman paljon resursseja puhtaan kvanttitietokonejärjestelmän toteuttamiseen vaan tärkeämpää lienee rakentaa kvanttialijärjestelmiä, joissa klassisen tietokoneen CPU:n, FPU:n ja GPU:n rinnalla on QPU ("kvanttisuoritin"). Todennäköisesti QPU:t ovat ainakin aluksi niin suurikokoisia, että yhteys tavallisen tietokoneen ja kvanttitietokoneen välillä hoituu Internetin ja pilvipalveluiden kautta.
- On tärkeää pyrkiä rakentamaan reversiibeleitä tietokonepiirejä, koska niissä on potentiaalia saada piirien energiankulutus haltuun. Kvanttitietokoneet olisivat joka tapauksessa reversiibeleitä.
- On tärkeää tutkia analogista laskentaa ja ternääristä laskentaa, koska niissä voi olla paljon hyödyntämätöntä laskentapotentiaalia. Kvanttitietokoneillakin on mahdollista tehdä ternääristä laskentaa kutriteillä. Kvanttitietokoneet voivat myös olla joko analogisia tai digitaalisia.
- On tutkittava uusia elektroniikan peruskomponentteja kuten memristoria ja muita muistielementtejä. Näille voi löytyä sovelluksia esimerkiksi tekoälystä ja muusta koneoppimisesta. Myös kvanttitietokoneet voivat hyötyä näistä uusista komponenteista.
- On pyrittävä hyödyntämään "tavallisten" tietokoneiden käyttämätöntä laskentatehoa tieteelliseen laskentaan. Esimerkiksi älypuhelin voisi akun lataamisen aikana suorittaa proteiinien laskostumiseen liittyvää laskentaa.
- Kvanttifysiikka asettaa rajat: Vain kvanttiteknologia mahdollistaa tehokkaan salakuuntelun estämisen ja/tai havaitsemisen, koska se on sisäänrakennettuna kvanttifysiikan laeissa. Kvanttiteknologialla voi tuottaa aidosti satunnaisia satunnaislukuja; nykyään satunnaisluvut ovat vielä lähes poikkeuksetta pseudosatunnaislukuja.

Lähteet

A Preview of Bristlecone, Google's New Quantum Processor. https://web.archive.org/web/20210113224646/https://ai.googleblog.com/2018/03/a-preview-of-bristlecone-googles-new.html. Käytetty: 2021-01-29.

Aaronson, S. 2013. *Quantum Computing Since Democritus.* Quantum Computing Since Democritus. Cambridge University Press. ISBN: 9780521199568. https://books.google.fi/books?id=jRGfhSoFx0oC.

Aaronson, Scott. 2008. "The limits of quantum". *Scientific American* 298 (3): 62–69.

Amico, Mirko, Zain H. Saleem ja Muir Kumph. 2019. "Experimental study of Shor's factoring algorithm using the IBM Q Experience". *Phys. Rev. A* 100 (1): 012305. https://doi.org/10.1103/PhysRevA.100.012305. https://link.aps.org/doi/10.1103/PhysRevA.100.012305.

Arute, Frank, Kunal Arya, Ryan Babbush, Dave Bacon, Joseph C Bardin, Rami Barends, Rupak Biswas, Sergio Boixo, Fernando GSL Brandao, David A Buell ym. 2019. "Quantum supremacy using a programmable superconducting processor". *Nature* 574 (7779): 505–510.

Atomico. *Our Investment in PsiQuantum: building the world's first useful quantum computer.* https://web.archive.org/web/20201005221127/https://medium.com/atomico/our-partnership-with-psiquantum-building-the-worlds-first-useful-quantum-computer-ac547b6b0a86. Käytetty: 2021-01-28.

Balla, P. C., ja A. Antoniou. 1984. "Low power dissipation MOS ternary logic family". *IEEE Journal of Solid-State Circuits* 19, numero 5 (lokakuu): 739–749. ISSN: 1558-173X. https://doi.org/10.1109/JSSC.1984.1052216.

Barmes, Itan, ja Bram Bosch. *Quantum computers and the Bitcoin blockchain - An analysis of the impact quantum computers might have on the Bitcoin blockchain.* https://web.archive.org/web/20210121195645/https://www2.deloitte.com/nl/nl/pages/innovatie/artikelen/quantum-computers-and-the-bitcoin-blockchain.html. Käytetty: 2021-01-22.

Benioff, Paul. 1980. “The computer as a physical system: A microscopic quantum mechanical Hamiltonian model of computers as represented by Turing machines”. *Journal of statistical physics* 22 (5): 563–591.

Bennett, C. H. 1973. “Logical Reversibility of Computation”. *IBM Journal of Research and Development* 17, numero 6 (marraskuu): 525–532. ISSN: 0018-8646. https://doi.org/10.1147/rd.176.0525.

Bernhardt, Chris. 2019. *Quantum computing for everyone.* Mit Press.

Bernstein, Daniel J, Johannes Buchmann ja Erik Dahmen. *Post-Quantum Cryptography.–2009.*

Bernstein, Ethan, ja Umesh Vazirani. 1997. “Quantum Complexity Theory”. *SIAM Journal on Computing* 26 (5): 1411–1473. https://doi.org/10.1137/S0097539796300921. eprint: https://doi.org/10.1137/S0097539796300921. https://doi.org/10.1137/S0097539796300921.

Bernstein-Vazirani Algorithm - A university quantum algorithms/computation course supplement based on Qiskit. https://web.archive.org/web/20201231022238/https://qiskit.org/textbook/ch-algorithms/bernstein-vazirani.html. Käytetty: 2021-01-19.

Bernstein-Vazirani Algorithm — Programming on Quantum Computers Season 1 Ep 6. https://www.youtube.com/watch?v=sqJIpHYl7oo. Käytetty: 2021-01-19.

Bitcoin Is Not Quantum-Safe, And How We Can Fix It When Needed. https://web.archive.org/web/20210110225040/https://bitcoinmagazine.com/articles/bitcoin-is-not-quantum-safe-and-how-we-can-fix-1375242150. Käytetty: 2021-01-11.

Bitcoin mining is NP-hard. https://web.archive.org/web/20210111224831/https://freedom-to-tinker.com/2014/10/27/bitcoin-mining-is-np-hard/. Käytetty: 2021-01-12.

Bitcoin vs. The NSA’s Quantum Computer. https://web.archive.org/web/20210110225036/https://www.bitcoinnotbombs.com/bitcoin-vs-the-nsas-quantum-computer/. Käytetty: 2021-01-11.

Bitcoin, kvanttitietokoneet ja kyberturvallisuus. https://web.archive.org/web/20210110225046/http://users.jyu.fi/~hetahein/lamport.html. Käytetty: 2021-01-11.

Brown, Julian, ja Kimmo Pietiläinen. 2001. *Kvanttitietokone.* Helsinki: Terra Cognita. ISBN: 9525202429, 9789525202427.

Buchholz, Scott, Joe Mariani, Adam Routh, Akash Keyal ja Pankaj Kishnani. *The realist's guide to quantum technology and national security - What nontechnical government leaders can do today to be ready for tomorrow's quantum world.* https://web.archive.org/web/20210121195544if_/https://www2.deloitte.com/us/en/insights/industry/public-sector/the-impact-of-quantum-technology-on-national-security.html/. Käytetty: 2021-01-21.

Cambou, Bertrand, Paul G. Flikkema, James Palmer, Donald Telesca ja Christopher Philabaum. 2018. "Can Ternary Computing Improve Information Assurance?" *Cryptography* 2 (1). ISSN: 2410-387X. https://doi.org/10.3390/cryptography2010006. https://www.mdpi.com/2410-387X/2/1/6.

Caraiman, Simona, ja Vasile Manta. 2013. "Image representation and processing using ternary quantum computing". Teoksessa *International Conference on Adaptive and Natural Computing Algorithms,* 366–375. Springer.

Caravelli, Francesco, ja Juan Pablo Carbajal. 2018. "Memristors for the Curious Outsiders". *Technologies* 6 (4). ISSN: 2227-7080. https://doi.org/10.3390/technologies6040118. https://www.mdpi.com/2227-7080/6/4/118.

Chapman, Peter. *Introducing the World's Most Powerful Quantum Computer.* https://web.archive.org/web/20201115070552/https://ionq.com/posts/october-01-2020-most-powerful-quantum-computer. Käytetty: 2021-01-28.

Chua, L. 1971. "Memristor-The missing circuit element". *IEEE Transactions on Circuit Theory* 18 (5): 507–519. https://doi.org/10.1109/TCT.1971.1083337.

Cirq - Wikipedia. https://web.archive.org/web/20210122190848/https://en.wikipedia.org/w/index.php?title=Cirq&oldid=975623955. Käytetty: 2021-01-22.

Connelly, Jeff. 2008. "Ternary computing testbed: 3-trit computer architecture". Tohtorinväitöskirja, California Polytechnic State University.

Conover, Emily. *The new light-based quantum computer Jiuzhang has achieved quantum supremacy.* https://web.archive.org/web/20210128173411/https://www.sciencenews.org/article/new-light-based-quantum-computer-jiuzhang-supremacy. Käytetty: 2021-01-28.

D-Wave Defies World of Critics With 'First Quantum Cloud'. https://web.archive.org/web/20210128232251/https://www.wired.com/2012/02/dwave-quantum-cloud/. Käytetty: 2021-01-29.

Denning, Dorothy E. 2019. "Is quantum computing a cybersecurity threat?" *American Scientist* 107 (2): 83–85.

Deutsch, David. 1985. "Quantum theory, the Church–Turing principle and the universal quantum computer". *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences* 400 (1818): 97–117.

Deutsch, David, ja Richard Jozsa. 1992. "Rapid solution of problems by quantum computation". *Proceedings of the Royal Society of London. Series A: Mathematical and Physical Sciences* 439 (1907): 553–558. https://doi.org/10.1098/rspa.1992.0167. eprint: https://royalsocietypublishing.org/doi/pdf/10.1098/rspa.1992.0167. https://royalsocietypublishing.org/doi/abs/10.1098/rspa.1992.0167.

Deutsch-Jozsa Algorithm - A university quantum algorithms/computation course supplement based on Qiskit. https://web.archive.org/web/20210106215122/https://qiskit.org/textbook/ch-algorithms/deutsch-jozsa.html. Käytetty: 2021-01-08.

DeWitt, B.S., H. Everett ja N. Graham. 1973. *The Many-worlds Interpretation of Quantum Mechanics.* Princeton series in physics. Princeton University Press. ISBN: 9780691081311. https://books.google.fi/books?id=SK13QgAACAAJ.

Di Ventra, M., Y. V. Pershin ja L. O. Chua. 2009. "Circuit Elements With Memory: Memristors, Memcapacitors, and Meminductors". *Proceedings of the IEEE* 97 (10): 1717–1724. https://doi.org/10.1109/JPROC.2009.2021077.

Ding, Jintai, ja Rainer Steinwandt. 2019. *Post-Quantum Cryptography: 10th International Conference, PQCrypto 2019, Chongqing, China, May 8–10, 2019 Revised Selected Papers.* Nide 11505. Springer.

Ding, Jintai, ja Jean-Pierre Tillich. 2020. *Post-Quantum Cryptography: 11th International Conference, PQCrypto 2020, Paris, France, April 15–17, 2020, Proceedings.* Nide 12100. Springer Nature.

Docs and Resources - IBM Quantum Experience - Deutsch-Jozsa algorithm. https://web.archive.org/web/20210106215139if_/https://quantum-computing.ibm.com/docs/iqx/guide/deutsch-jozsa-algorithm. Käytetty: 2021-01-08.

Docs and Resources - IBM Quantum Experience - Entanglement. https://web.archive.org/web/20210103222424if_/https://quantum-computing.ibm.com/docs/iqx/guide/entanglement. Käytetty: 2021-01-04.

Docs and Resources - IBM Quantum Experience - Shor's algorithm. https://web.archive.org/web/20201101072900/https://quantum-computing.ibm.com/docs/iqx/guide/shors-algorithm. Käytetty: 2020-12-29.

Dwave Systems shows off quantum chip with 2048 physical qubits. https://web.archive.org/web/20150513044844/http://nextbigfuture.com/2014/10/dwave-systems-shows-off-quantum-chip.html. Käytetty: 2021-01-29.

Dwork, Cynthia, ja Moni Naor. 1993. "Pricing via Processing or Combatting Junk Mail". Teoksessa *Advances in Cryptology — CRYPTO' 92,* toimittanut Ernest F. Brickell, 139–147. Berlin, Heidelberg: Springer Berlin Heidelberg. ISBN: 978-3-540-48071-6.

Dyakonov, Mikhail I. 2020. *Will We Ever Have a Quantum Computer?* Springer.

Example program to demonstrate BB84 QKD Protocol. https://web.archive.org/web/20210108144928/https://raw.githubusercontent.com/quantumlib/Cirq/master/examples/bb84.py. Käytetty: 2021-01-09.

Feynman, Richard P. 1982. "Simulating physics with computers". *Int. J. Theor. Phys* 21 (6/7).

Frank, Michael P. 2006. *Nanocomputer systems engineering.* CRC Press.

Future quantum computers may have to be built UNDERGROUND. https://web.archive.org/web/20210203143445/https://bigworldtale.com/science/future-quantum-computers-may-have-to-be-built-underground/. Käytetty: 2021-02-03.

Generating a Bitcoin Address with JavaScript. https://web.archive.org/web/20201111202407/http://procbits.com/2013/08/27/generating-a-bitcoin-address-with-javascript. Käytetty: 2021-01-19.

Hardware - Xanadu. https://web.archive.org/web/20210128191607/https://xanadu.ai/hardware. Käytetty: 2021-01-28.

Hayes, Brian. 2014. "Programming your quantum computer". *American Scientist* 102 (1): 22–25.

Hidary, J.D. 2019. *Quantum Computing: An Applied Approach.* Springer International Publishing. ISBN: 9783030239220. https://books.google.fi/books?id=nymsDwAAQBAJ.

Hirvensalo, M. 2003. *Quantum Computing.* Natural Computing Series. Springer Berlin Heidelberg. ISBN: 9783540407041. https://books.google.fi/books?id=oCd5fWPqf7UC.

IBM Quantum systems overview. https://web.archive.org/web/20201205143751if_/https://quantum-computing.ibm.com/docs/manage/backends/. Käytetty: 2020-12-05.

Insights, CB. 2020. *2020 Tech Trends.* Tekninen raportti. CB Insights.

Johnston, E.R., N. Harrigan ja M. Gimeno-Segovia. 2019. *Programming Quantum Computers: Essential Algorithms and Code Samples.* O'Reilly Media. ISBN: 9781492039655. https://books.google.fi/books?id=QqegDwAAQBAJ.

Kaiser, S.C., ja C. Granade. 2021. *Learn Quantum Computing with Python and Q#: A hands-on approach.* Manning Publications. ISBN: 9781617296130. https://books.google.fi/books?id=BPZ0zQEACAAJ.

Kerntopf, Pawel. *Reversible Logic Circuits.* https://web.archive.org/web/20180712185116/http://web.cecs.pdx.edu/~mperkows/CLASS_VHDL_99/tran888/lecture003-reversible-logic.pdf. Käytetty: 2021-01-22.

Killoran, Nathan, Josh Izaac, Nicolás Quesada, Ville Bergholm, Matthew Amy ja Christian Weedbrook. 2019. "Strawberry Fields: A Software Platform for Photonic Quantum Computing". *Quantum* 3 (maaliskuu): 129. ISSN: 2521-327X. https://doi.org/10.22331/q-2019-03-11-129. https://doi.org/10.22331/q-2019-03-11-129.

Krishnakumar, A. 2020. *Quantum Computing and Blockchain in Business: Exploring the applications, challenges, and collision of quantum computing and blockchain.* Packt Publishing. ISBN: 9781838646134. https://books.google.fi/books?id=gNDaDwAAQBAJ.

Lamport, Leslie. 1979. *Constructing digital signatures from a one-way function.* Tekninen raportti. Citeseer.

Landauer, Rolf. 1961. "Irreversibility and heat generation in the computing process". *IBM journal of research and development* 5 (3): 183–191.

List of quantum processors - Wikipedia. https://web.archive.org/web/20210113205014/https://en.wikipedia.org/w/index.php?title=List_of_quantum_processors&oldid=999209651. Käytetty: 2021-01-13.

Loredo, R. 2020. *Learn Quantum Computing with Python and IBM Quantum Experience: A hands-on introduction to quantum computing and writing your own quantum programs with Python.* Packt Publishing. ISBN: 9781838986759. https://books.google.fi/books?id=ufz%5C_DwAAQBAJ.

Lu, Dawei, Boruo Xu, Nanyang Xu, Zhaokai Li, Hongwei Chen, Xinhua Peng, Ruixue Xu ja Jiangfeng Du. 2012. "Quantum chemistry simulation on quantum computers: theories and experiments". *Phys. Chem. Chem. Phys.* 14 (26): 9411–9420. https://doi.org/10.1039/C2CP23700H. http://dx.doi.org/10.1039/C2CP23700H.

Manin, Yu I. 1980. *Vychislimoe i nevychislimoe (Computable and Noncomputable), Moscow: Sov.*

Many-Worlds Interpretation of Quantum Mechanics. http://web.archive.org/web/20210108082405/https://plato.stanford.edu/entries/qm-manyworlds/. Käytetty: 2021-01-12.

Martin-Lopez, Enrique, Anthony Laing, Thomas Lawson, Roberto Alvarez, Xiao-Qi Zhou ja Jeremy L O'brien. 2012. "Experimental realization of Shor's quantum factoring algorithm using qubit recycling". *Nature photonics* 6 (11): 773–776.

Ménard, Alexandre, Ivan Ostojic, Mark Patel ja Daniel Volz. *Quantum algorithm breakthrough.* https://web.archive.org/web/20210121195055/https://www.mckinsey.com/business-functions/mckinsey-digital/our-insights/a-game-plan-for-quantum-computing#. Käytetty: 2021-01-21.

Metodi, T.S., A.I. Faruque ja F.T. Chong. 2011. *Quantum Computing for Computer Architects: Second Edition.* Synthesis Lectures on Computer Architecture. Morgan & Claypool Publishers. ISBN: 9781608456215. https://books.google.fi/books?id=QZBgAQAAQBAJ.

Moor Insights and Strategy ja Smith-Goodson, Paul. *Xanadu Brings Photonic Quantum Computing To The Cloud.* https://web.archive.org/web/20210128190249/https://www.forbes.com/sites/moorinsights/2020/09/09/xanadu-brings-photonic-quantum-computing-to-the-cloud/?sh=508ba58864e6. Käytetty: 2021-01-28.

Moran, C.C. 2019. *Mastering Quantum Computing with IBM QX: Explore the World of Quantum Computing Using the Quantum Composer and Qiskit.* Packt Publishing. ISBN: 9781789136432. https://books.google.fi/books?id=I6lPuwEACAAJ.

Multiple Qubits and Entangled States - A university quantum algorithms/computation course supplement based on Qiskit. https://web.archive.org/web/20210119202145/https://qiskit.org/textbook/ch-gates/multiple-qubits-entangled-states.html. Käytetty: 2021-01-19.

National Academies of Sciences, Engineering, Medicine ym. 2019. *Quantum computing: progress and prospects.* National Academies Press.

Neittaanmaki, Pekka, ja Anthony Ogbechie. *Quantum Computing and digital Security.* Käytetty: 2021-01-21.

———. *QUANTUM COMPUTING COURSES, DEGREES AND RESEARCH GROUPS.* Käytetty: 2021-01-21.

———. *Quantum Computing trends 2020 and beyond - Expert opinions and forecasts.* Käytetty: 2021-01-21.

———. *QUANTUM COMPUTING: ALGORITHMS AND SOFTWARE TRENDS.* Käytetty: 2021-01-21.

New York, City College of. *Quantum algorithm breakthrough.* https://web.archive.org/web/20210121195246/www.sciencedaily.com/releases/2020/11/201116092255.htm. Käytetty: 2021-01-21.

Nielsen, M.A., ja I.L. Chuang. 2010. *Quantum Computation and Quantum Information: 10th Anniversary Edition.* Cambridge University Press. ISBN: 9781139495486. https://books.google.fi/books?id=-s4DEy7o-a0C.

Norlén, H. 2020. *Quantum Computing in Practice with Qiskit® and IBM Quantum Experience®: Practical recipes for quantum computer coding at the gate and algorithm level with Python.* Packt Publishing. ISBN: 9781838821036. https://books.google.fi/books?id=pQ4LEAAAQBAJ.

OECD. 2020. *Artificial intelligence, blockchain and quantum computing.* https://doi.org/https://doi.org/https://doi.org/10.1787/c51bcfeb-en. https://www.oecd-ilibrary.org/content/component/c51bcfeb-en.

Pednault, Edwin, John Gunnels, Dmitri Maslov ja Jay Gambetta. *On "Quantum Supremacy" | IBM Research Blog.* https://web.archive.org/web/20210121195520/https://www.ibm.com/blogs/research/2019/10/on-quantum-supremacy/. Käytetty: 2021-01-21.

Peotta, Sebastiano, ja Massimiliano Di Ventra. 2014. "Superconducting Memristors". *Physical Review Applied* 2, numero 3 (syyskuu). ISSN: 2331-7019. https://doi.org/10.1103/physrevapplied.2.034011. http://dx.doi.org/10.1103/PhysRevApplied.2.034011.

Porat, DI. 1969. "Three-valued digital systems". Teoksessa *Proceedings of the Institution of Electrical Engineers,* 116:947–954. 6. IET.

Proos, John, ja Christof Zalka. 2003. "Shor's discrete logarithm quantum algorithm for elliptic curves". *arXiv preprint quant-ph/0301141.*

Quantum computing in a nutshell - Qiskit 0.23.2 documentation. https://web.archive.org/web/20210104185724/https://qiskit.org/documentation/qc_intro.html. Käytetty: 2021-01-04.

Quantum information - Wikipedia. https://web.archive.org/web/20210112190417/https://en.wikipedia.org/w/index.php?title=Quantum_information&oldid=997558071#Qubits_and_quantum_information. Käytetty: 2021-01-12.

Quantum Supremacy Using a Programmable Superconducting Processor. https://web.archive.org/web/20210121195621/https://ai.googleblog.com/2019/10/quantum-supremacy-using-programmable.html. Käytetty: 2021-01-29.

Quantum Volume – Qiskit 0.23.3 documentation. https://web.archive.org/web/20210128163919/https://qiskit.org/documentation/tutorials/noise/5_quantum_volume.html. Käytetty: 2021-01-28.

Quantum Volume: The Power of Quantum Computers. https://web.archive.org/web/20210128192159/https://www.honeywell.com/us/en/news/2020/03/quantum-volume-the-power-of-quantum-computers. Käytetty: 2021-01-28.

Al-Rabadi, Anas N. 2004. *Reversible logic synthesis: From fundamentals to quantum computing.* Springer Science & Business Media.

Radwan, Ahmed G, ja Mohammed E Fouda. 2015. *On the mathematical modeling of memristor, memcapacitor, and meminductor.* Nide 26. Springer.

Rahmani, Armin, Kevin J. Sung, Harald Putterman, Pedram Roushan, Pouyan Ghaemi ja Zhang Jiang. 2020. "Creating and Manipulating a Laughlin-Type $\nu = 1/3$ Fractional Quantum Hall State on a Quantum Computer with Linear Depth Circuits". *PRX Quantum* 1 (2): 020309. https://doi.org/10.1103/PRXQuantum.1.020309. https://link.aps.org/doi/10.1103/PRXQuantum.1.020309.

Rieffel, E., ja W. Polak. 2014. *Quantum Computing: A Gentle Introduction.* Scientific and Engineering Computation. MIT Press. ISBN: 9780262526678. https://books.google.fi/books?id=jL34DwAAQBAJ.

Run Grover's search algorithm in Q - Quantum Development Kit - Microsoft Quantum | Microsoft Docs. https://web.archive.org/web/20210108142834/https://docs.microsoft.com/en-us/quantum/tutorials/search. Käytetty: 2021-01-11.

Scalable quantum systems built from the chip up to power practical applications. https://web.archive.org/web/20210128214700if_/https://www.rigetti.com/what. Käytetty: 2021-01-28.

Schuld, Maria, ja Francesco Petruccione. 2018. *Supervised learning with quantum computers.* Springer.

Shor, P. W. 1994. "Algorithms for quantum computation: discrete logarithms and factoring". Teoksessa *Proceedings 35th Annual Symposium on Foundations of Computer Science,* 124–134. Marraskuu. https://doi.org/10.1109/SFCS.1994.365700.

Silva, V. 2018. *Practical Quantum Computing for Developers: Programming Quantum Rigs in the Cloud using Python, Quantum Assembly Language and IBM QExperience.* Apress. ISBN: 9781484242186. https://books.google.fi/books?id=Scl-DwAAQBAJ.

Smith, Kenneth C. 1981. "The prospects for multivalued logic: A technology and applications view". *IEEE Transactions on Computers,* numero 9, 619–634.

Sutor, R.S. 2019. *Dancing with Qubits: How quantum computing works and how it can change the world.* Packt Publishing. ISBN: 9781838825256. https://books.google.fi/books?id=NA3UDwAAQBAJ.

The qubit - Docs and Resources - IBM Quantum Experience. https://web.archive.org/web/20210128164055if_/https://quantum-computing.ibm.com/docs/iqx/guide/the-qubit. Käytetty: 2021-01-28.

The Reversible Computing FAQ (Frequently Asked Questions). https://web.archive.org/web/20200812125914/https://www.cise.ufl.edu/research/revcomp/faq.html. Käytetty: 2021-01-19.

Toffoli, Tommaso. 1980. "Reversible computing". Teoksessa *International Colloquium on Automata, Languages, and Programming,* 632–644. Springer.

Tsividis, Yannis. 2018. "Not your father's analog computer". *IEEE Spectrum* 55 (2): 38–43.

Vasara, Antti ja Majumdar, Himadri. *Suomen ensimmäisen kvanttitietokoneen rakentaminen alkaa – VTT:n kumppanina suomalainen IQM.* https://web.archive.org/web/20210128194630/https://www.vttresearch.com/fi/uutiset-ja-tarinat/suomen-ensimmaisen-kvanttitietokoneen-rakentaminen-alkaa-vttn-kumppanina. Käytetty: 2021-01-28.

Ventra, Massimiliano Di, ja Yuriy V Pershin. 2013. "On the physical properties of memristive, memcapacitive and meminductive systems". *Nanotechnology* 24, numero 25 (toukokuu): 255201. https://doi.org/10.1088/0957-4484/24/25/255201. https://doi.org/10.1088%2F0957-4484%2F24%2F25%2F255201.

Vepsäläinen, Antti P, Amir H Karamlou, John L Orrell, Akshunna S Dogra, Ben Loer, Francisca Vasconcelos, David K Kim, Alexander J Melville, Bethany M Niedzielski, Jonilyn L Yoder ym. 2020. "Impact of ionizing radiation on superconducting qubit coherence". *Nature* 584 (7822): 551–556.

What are the Q programming language and QDK? https://web.archive.org/web/20201205145652/https://docs.microsoft.com/en-us/quantum/overview/what-is-qsharp-and-qdk. Käytetty: 2020-12-05.

Vos, J. 2021. *Quantum Computing in Action.* Manning Publications. ISBN: 9781617296321. https://books.google.fi/books?id=tz3MyQEACAAJ.

Yanofsky, N.S., ja M.A. Mannucci. 2008. *Quantum Computing for Computer Scientists.* Cambridge University Press. ISBN: 9781139643900. https://books.google.fi/books?id=U1chAwAAQBAJ.

Zhong, Han-Sen, Hui Wang, Yu-Hao Deng, Ming-Cheng Chen, Li-Chao Peng, Yi-Han Luo, Jian Qin ym. 2020. "Quantum computational advantage using photons". *Science* 370 (6523): 1460–1463. ISSN: 0036-8075. https://doi.org/10.1126/science.abe8770. eprint: https://science.sciencemag.org/content/370/6523/1460.full.pdf. https://science.sciencemag.org/content/370/6523/1460.

Liitteet

A Liite: Kvanttilaskennan trendit vuodelle 2020 ja eteenpäin

Tämä liite sisältää tietoa kvanttilaskennan trendeistä vuodelle 2020 ja pitemmälle. Se on lyhyt yhteenveto professori Pekka Neittaanmäen ja Anthony Ogbechien materiaalista "Quantum Computing Trends 2020 And Beyond" (Neittaanmaki ja Ogbechie).

A.1 Tekoäly, lohkoketjut ja kvanttilaskenta

OECD:n digitaalisen ekonomian näkemyksen mukaan kesäkuussa 2020 yli 60 valtiolla oli kansallinen strategia tai ohjeistuksia liittyen tekoälyyn. Monet maat kehittävät myös hajautetun tilikirjan teknologioita, jotka usein tunnetaan paremmin lohkoketjuina. Kolmas kiinnostava teknologiatrendi on kvanttilaskenta. Yhdysvallat hallitsee kvanttilaskennan, Eurooppa kvanttimekaniikan (fysiikan näkökulmasta) ja Kiina kvanttiviestinnän ja kvanttikryptografian. (OECD 2020)

A.2 Kvanttitietokone tekee jotain uutta

Vuonna 2019 Googlen tutkijat ilmoittivat saavuttaneensa kvanttiherruuden eli tilanteen, jossa kvanttitietokone tekee ensimmäistä kertaa jotain sellaista, johon klassinen tietokone ei kykene. Googlen Sycamore-niminen 53-kubittinen kvanttitietokone käytti 200 sekuntia laskuoperaatioon, joka olisi vienyt maailman nopeimmalta klassiselta supertietokoneelta 10 000 vuotta. (Arute ym. 2019; Pednault ym.)

A.3 Kyberturvallisuudesta

Kvanttiavainten jakaminen (QKD) näyttelee tärkeää roolia kyberturvallisuuden työkaluna CB Insightsin 2020 Tech Trends -raportin mukaan. (Insights 2020) Kvanttiavainten jakamista käsitellään myös tämän raportin sivulla 16.

Lisäksi materiaalista (Neittaanmaki ja Ogbechie) löytyy viitteitä esimerkiksi Deloitten (Buch-

holz ym.) ja McKinseyn (Ménard ym.) uutisartikkeleihin kvanttilaskennasta kyberturvallisuuden näkökulmasta. McKinseyn artikkelin mukaan aikavälillä 2030–2040 saattaa olla olemassa kvanttitietokone, joka hallitsee satoja tuhansia tai jopa miljoonia kubitteja, ja kykenee suorittamaan Shorin algoritmia mielekkäästi. Tuossa vaiheessa tarvitaan jo kvanttikryptaustekniikoita.

B Liite: Kvanttialgoritmit ja kvanttisovellukset

Tämä liite sisältää tietoa kvanttilaskennan algoritmeista ja sovelluksista. Se on lyhyt yhteenveto professori Pekka Neittaanmäen ja Anthony Ogbechien materiaalista "Quantum Algorithms and Applications" (Neittaanmaki ja Ogbechie).

B.1 Kyberturvallisuudesta

Laajennetun Churchin–Turingin teesin (National Academies of Sciences, Medicine ym. 2019) mukaan laskennallisen ongelman ratkaisun nopeuttamiseksi pitää

1. vähentää aikaa yhden askeleen käymiseen
2. ottaa monia askeleita rinnakkain
3. vähentää askeleitten kokonaismäärää suunnittelemalla älykäs algoritmi.

On havaittu, että kvanttitietokoneet ratkaisevat tiettyjä tehtäviä eksponentiaalisesti vähemmin askelin kuin parhaat klassiset algoritmit. Peter Shorin kvanttialgoritmit jakavat suuria lukuja tekijöihinsä ja laskevat diskreettejä logaritmeja eksponentiaalisesti nopeammin kuin algoritmit klassisille tietokoneille. Nämä kvanttialgoritmit aiheuttivat suurta huolta tietoturvayhteisöissä, koska suurin osa yhteiskunnan digitaalisesta tiedosta on suojattuna "klassisesti kovien" julkisten avainten kryptojärjestelmiin. (National Academies of Sciences, Medicine ym. 2019)

Kyberturvallisuus siis on avainasemassa kvanttilaskentajärjestelmissä.

Shorin algoritmia käsitellään enemmän tämän raportin sivulla 24.

B.2 Kvanttifysiikan simulointi kvanttitietokoneella

Toisena esimerkkinä mainittakoon uutinen läpimurrosta kvanttifysiikan simuloinnissa (New York), jonka mukaan fyysikot ovat löytäneet kvanttialgoritmin, jonka avulla on potentiaalia tutkia monen elektronin kvanttijärjestelmiä. (Rahmani ym. 2020)

C Liite: Kvanttilaskennan kurssit, tutkinnot ja tutkimusryhmät

Tämä liite sisältää tietoa kvanttilaskentaan liittyvästä opetuksesta, tutkinnoista ja tutkimusryhmistä. Se on lyhyt yhteenveto professori Pekka Neittaanmäen ja Anthony Ogbechien materiaalista "Quantum Computing Courses, Degrees and Research Groups" (Neittaanmaki ja Ogbechie).

C.1 Johtavia tutkimuslaitoksia kvanttilaskennassa

Johtavia yliopistoja kvanttilaskentatutkimuksessa ovat Waterloon yliopisto, Oxfordin yliopisto, Harvardin yliopisto, MIT, Kvanttiteknologioiden keskus Singaporessa, Berkeleyn yliopisto, Marylandin yliopisto, Kiinan tiede- ja teknologiayliopisto, Chicagon yliopisto, Sydneyn yliopisto, QAR-Lab Münchenissä ja Innsbruckin yliopisto.

C.2 Kurssitarjontaa lähiopetuksena ja etäopetuksena

Internetistä löytyy runsaasti sekä maksullisia että maksuttomia kursseja kvanttimekaniikkaan ja kvanttilaskentaan liittyen. Kursseja ja tutkintoja järjestetään toki myös läsnäopetuksena. *Suomessa kvanttilaskentatutkimusta ja -opetusta järjestetään ainakin Aalto-yliopistolla, Turun yliopistolla sekä Jyväskylän yliopiston IT-tiedekunnassa ja fysiikan laitoksella.* Ohessa on listattuna kahdeksan nettipohjaista kvanttitietokonekurssia esimerkin vuoksi.

1. "UC Berkeley's Quantum Mechanics and Quantum Computation on edX" https://www.edx.org/course/quantum-mechanics-and-quantum-computation
2. "DelftX The Quantum Internet and Quantum Computers: How Will They Change the World?"
   https://www.edx.org/course/the-quantum-internet-and-quantum-computers-how-w-2

3. “DelftX Professional Certificate Quantum Computing and Quantum Internet” https://www.edx.org/professional-certificate/delftx-quantum-computing-and-quantum-internet
4. “MITxPro Professional Certificate in Quantum Computing” https://learn-xpro.mit.edu/quantum-computing
5. “MIT Quantum Information Science I from edX” https://www.edx.org/course/quantum-information-science-i-part-1
6. “Purdue Micromasters Introduction to Quantum Technology from edX” https://www.edx.org/course/introduction-to-quantum-technology
7. “Understanding Quantum Computers from Keio University on FutureLearn” https://www.classcentral.com/course/intro-to-quantum-computing-8706
8. “Russian language Quantum Computing Course from Coursera” https://www.coursera.org/learn/kvantovyye-vychisleniya

D Liite: Kvanttilaskenta ja digitaalinen turvallisuus

Tämä liite sisältää tietoa kvanttilaskentaan liittyvästä digitaalisesta turvallisuudesta. Puhutaan myös kyberturvallisuudesta. Se on lyhyt yhteenveto professori Pekka Neittaanmäen ja Anthony Ogbechien materiaalista “Quantum Computing and Digital Security” (Neittaanmaki ja Ogbechie).

D.1 Kryptovaluutat turvaan kvanttitietokonehyökkäyksiltä

Kuten tämän raportin sivulla 30 jo todettiin, Bitcoin voi olla vaarassa kvanttitietokoneaikakaudella, mutta onneksi on olemassa keinoja tehdä Bitcoinista kvanttitietokoneturvallinen. Deloitten artikkelin (Barmes ja Bosch) mukaan kvanttitietokoneturvallisuuden suhteen on eroa siinä, minkä tyyppisiä bitcoin-osoitteita käytetään bitcoin-kolikoiden säilyttämiseen. Bitcoinin alkuvuosina käytettiin p2pk-tyyppisiä (engl. “pay to public key”, suom. “maksa julkiselle avaimelle”) bitcoin-osoitteita, joissa ei ollut esimerkiksi uudemmista bitcoin-osoitteista tuttua tarkistussummaa, joka tehokkaasti estää virhelyönnit osoitteista, jolloin bitcoineja ei vahingossa katoa olemattomiin osoitteisiin.

Toinen ongelma on osoitteiden suuri pituus, joka johtaa tietokooltaan suurempiin transak-

tioihin ja pidempiin prosessointiaikoihin. Kvanttitietokoneturvallisuuden suhteen ongelma on siinä, että julkinen avain on suoraan saatavissa bitcoin-osoitteesta. Koska transaktiot ovat aina Bitcoinissa julkisia, voi kuka tahansa saada julkisen avaimen p2pk-osoitteesta. Kvanttitietokone, joka ajaa Shorin algoritmia, voisi siten etsiä yksityisen avaimen tämän julkisen avaimen/osoitteen perusteella. Yksityisen avaimen löydyttyä, ovat kyseisen avaimen osoitteeseen kuuluvat bitcoin-kolikot käytettävissä.

Saman artikkelin mukaan uudempi bitcoin-osoitetyyppi on p2pkh-tyyppinen (engl. "pay to public key hash", suom. "maksa julkisen avaimen tiivisteelle"), joka kehitettiin alun perin ratkaisemaan bitcoinien katoamisongelma virhelyöntien seurauksena (keksittiin alkaa käyttää tarkistussummaa) sekä osoitteen pituusongelma. Julkista avainta ei voi enää saada p2pkh-tyyppisestä bitcoin-osoitteesta. Julkinen avain paljastuu nyt vain, kun osoitteesta siirretään bitcoin-kolikoita muualle eli tehdään transaktio. Siispä, bitcoin-kolikot ovat turvassa kvanttitietokoneilta, kunhan p2pkh-osoitetta ei ole koskaan käytetty missään transaktiossa. Tästä syystä ei suositella bitcoin-osoitteita uudelleenkäytettäviksi, koska siinä tapauksessa osoitteen julkinen avain on jo aiemmin paljastunut. Kvanttitietokoneella voisi siis varastaa bitcoin-kolikoita kaikista p2pk-bitcoin-osoitteista ja uudelleenkäytetyistä p2pkh-bitcoin-osoitteista. Artikkelin mukaan noin 4 miljoonaa BTC, noin neljäsosa kaikista artikkelin kirjoitushetkellä olemassa olevista bitcoin-kolikoista, on vaarassa joutua kvanttitietokoneen haavoittamaksi.

D.2 Suojautumista kvanttitietokonehyökkäyksiltä

American Scientist -lehden artikkelissa (Denning 2019) mainitaan, että The National Academies -tutkimuksen mukaan tulevaisuuden koodinrikkovien kvanttitietokoneilla on oltava 100 000 kertaa niin paljon prosessointitehoa ja 100 kertaa parempi virhesuhde (engl. "error rate") kuin vuoden 2019 parhailla kvanttitietokoneilla. Saman tutkimuksen mukaan ei ole odotettavissa, että sellainen kvanttitietokone ilmaantuisi vielä 2020-luvun aikana.

Yksi suojautumiskeino Groverin algoritmia vastaan, artikkelin mukaan, on avainten koon pidentäminen: 256-bittinen avain on yhtä turvallinen kvanttitietokoneita vastaan kuin 128-bittinen avain on "tavallisia" hyökkäyksiä vastaan.

Shorin algoritmeja vastaan suojautuminen on ongelmallisempaa. Tekniikat kuten RSA, Diffie–Hellman ja jopa elliptisen käyrän kryptografia (Proos ja Zalka 2003) ovat vaarassa.

Informaatioteknologian tiedekunnan julkaisuja
No. 88/2021

ISBN 978-951-39-8534-9 (verkkoj.)
ISSN 2323-5004

JYVÄSKYLÄN YLIOPISTO