En introduktion till kvantdatorer

Lars Engblom
Kandidatavhandling i datavetenskap
Handledare: Jan Westerholm
Fakulteten för naturvetenskaper och teknik
Åbo Akademi

Våren 2018

Referat

Efterhand som utvecklingen av både kvantdatorer och deras simulering går framåt, ser man allt oftare artiklar om dessa ämnen i nyhetsflödet. För att förstå artiklarna och för att kunna bedöma dem kritiskt behöver läsaren förstå principerna bakom kvantdatorer. Den här avhandlingen förklarar grundprinciperna och matematiken bakom dem.

Till dessa grunder hör en kort introduktion till kvantmekanik, vektorer, matriser och bra-ket-notation eller Diracnotation, som den också kallas. Byggstenarna för kvantalgoritmer är kvantgrindar och några av dessa kommer att behandlas. Slutligen kommer Deutsch–Jozsa-algoritmen att presenteras, som ett exempel på en kvantalgoritm.

Innehåll

1 Introduktion och syfte **3**

2 Kvantbitar **4**

2.1 Kvantmekaniken upptäcks 4

2.2 Kvantbitar och deras fysiska representation 5

2.3 Schrödingers katt och superposition 7

2.4 Sammanflätade kvantbitar 7

2.5 Matematisk representation 8

2.5.1 Kvantbitar är vektorer 8

2.5.2 Bra-ket-notationen / Diracnotationen 9

2.5.3 Matrisnotation 10

3 Kvantgrindar **12**

3.1 Kvantgrindar som matriser 12

3.2 NOT-grinden 13

3.3 Hadamard 14

3.4 CNOT 15

4 Deutsch-Jozsa-algoritmen **18**

5 Sammanfattning och slutsats **24**

1 Introduktion och syfte

Kvantdatorer skiljer sig märkbart från den klassiska datorn, både till konstruktion och funktion. En klassisk dator är till sin grund så enkel att en intresserad och kunnig entusiast skulle kunna bygga en programmerbar sådan helt från grunden i sitt hem. Kvantdatorer är motsatsen. Man är flera år från att ha en fungerande kvantdator och det pågår aktiv forskning i hur man praktiskt ska kunna konstruera en sådan.

Intresset för kvantdatorer ligger naturligtvis i att man kan göra en del beräkningar betydligt snabbare med en sådan, än med en klassisk dator. Den kanske mest kända tillämpningen är faktorisering av stora heltal, då detta gör det möjligt att knäcka RSA och därmed får man tillgång till krypterad information. RSA är en asymmetrisk krypteringsalgoritm med offentlig och privat nyckel baserad på produkten av två stora primtal. Om man kan genom faktorisering återskapa de två primfaktorerna, så har man den privata nyckeln med vilken man kan dekryptera meddelandet. Det finns ingen känd algoritm för klassiska datorer som skulle klara av att faktorisera så stora primtal tillräckligt snabbt för att vara användbar.

Styrkan hos en kvantdator ligger i kvantbitarna. Kvantbitarna kan vara i något som kallas för en superposition, och på det sättet representerar de flera värden på en gång. Deutsch-Jozsa-algoritmen, som också behandlas som en del av avhandlingen är ett exempel på hur en kvantdator behöver betydligt färre operationer än en klassisk dator.

För att snabba på forskning kommer EU att satsa ca 1 miljard euro på kvantteknologi genom projektet ”The European quantum technologies flagship programme”, som börjar 2019 [1]. Att EU satsar sådana summor ger en uppfattning om hur viktig och intressant denna teknologi är. Samtidigt ser man också i nyhetsflödet hur kvantdatorer och simuleringen av dem nämns allt oftare.

Denna avhandling riktar sig till alla som studerar datavetenskap och önskar få en grundförståelse om kvantdatorer. En sådan grundförståelse hjälper om man vill följa med nyheter och diskussioner rörande ämnet. Mitt syfte är att presentera de allra mest väsentliga grundprinciperna på ett enkelt sätt, utan att läsaren behöver djupa förkunskaper inom fysik och matematik.

2 Kvantbitar

Kvantbitar är kvantdatorns motsvarighet till den klassiska datorns bitar. För att förstå hur kvantbitar fungerar behöver man en uppfattning om vad kvantmekanik handlar om.

I avsnitt 2.1 finns en kort beskrivning av hur kvantmekaniken upptäcks tillsammans med två exempel på kvantmekaniska fenomen, nämligen energin i termisk strålning och elektronskalens energinivåer i Bohrs atommodell.

Därefter behandlas i avsnitt 2.2 hur kvantmekaniska fenomen kan utnyttjas för att skapa en kvantbit. Josephsonövergångar presenteras som ett konkret exempel på kvantfenomen som utnyttjas av bland annat IBM.

I avsnitt 2.3 presenteras superposition. Då en kvantbit är i en superposition, har den en viss sannolikhet för att vara i det ena eller det andra tillståndet då den observeras. Man säger att superpositionen kollapsar.

Sammanflätade kvantbitar presenteras i avsnitt 2.4. Då två kvantbitar är i ett sammanflätat tillstånd befinner de sig i superposition, och om den ena kvantbiten observeras, kollapsar båda kvantbitarna till motsatta tillstånd. De två sammanflätade kvantbitarna bildar ett system fram till att någondera observeras och de kan inte beskrivas var för sig.

Därefter ges i avsnitt 2.5 den matematiska representationen av kvantbitar. De är vektorer som kan beskrivas med både bra-ket-notation och med matriser.

2.1 Kvantmekaniken upptäcks

Liksom relativitetsteorin, är kvantmekaniken en rätt så ny teori. Idéerna utvecklades under början av 1900-talet.

Omkring 1860 upptäckte Kirchhoff att det finns ett samband mellan temperatur och frekvens, som tillsammans med emissivitet beskriver den termiska strålning som en kropp avger [2]. Om man värmer en svart kropp, det vill säga ett föremål som absorberar all elektromagnetisk strålning, kan man alltså bestämma dess temperatur enligt den strålning kroppen avger.

Det visade sig vara svårt att hitta en funktion som beskriver detta samband och först 1900 löste Max Planck problemet. Han införde en konstant, som har fått

namnet Plancks konstant[3]. Med hjälp av Plancks konstant får man funktionen $E = nhv$ där E är energin i strålningen, n är ett heltal, h är Plancks konstant och v är strålningens frekvens[4]. Eftersom Plancks konstant är en faktor i energimängden, kan energin bara anta diskreta värden, även då $n = 1$. Strålningsenergin från dessa kroppar är alltså kvantifierad.

Under 1900-talets första hälft upptäcktes flera kvantfenomen. Till exempel i Bohrs atommodell (1913) finns bestämda elektronskal[5]. När en atom belyses så kan en elektron hoppa från ett skal med en lägre energinivå till ett skal med en högre energinivå. Elektronen flyttas omedelbart, utan att någonsin vara mittemellan dessa skal. Alltså är här också en kvantifiering av energinivåerna.

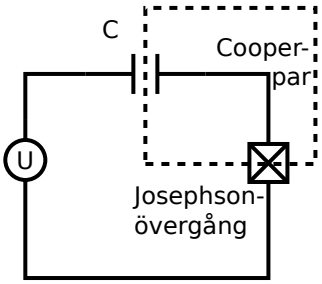
Trots att kvantmekaniken redan under 1900-talets första hälft blev en övergripande fysisk teori[6], tog det lång tid innan forskningen om kvantdatorer tog fart. Visst hade tanken om kvantbitar förekommit[7], men det var först 1980 som Jurij Manin på allvar presenterade idén om kvantdatorer[8][9]. Andra viktiga pionjär är Richard Feynman och David Deutsch[9]. Det är David Deutsch som står bakom grundidén till det problem som presenteras i avsnitt 4.

2.2 Kvantbitar och deras fysiska representation

Den klassiska datorn använder sig av bitar, som är antingen 0 eller 1. En bit är lätt att representera i elektronik; man kan använda två olika nivåer för spänning eller ström, där ena nivån representerar 0 och den andra nivån 1. Vidare är operationer med dessa bitar alltid deterministiska. Detta gör att man inte kan generera äkta slumptal med en klassisk dator utan specialgjord hårdvara för ändamålet.

Kvantbiten kan tyckas vara liknande till en början, eftersom allt som behövs är att man väljer två diskreta tillstånd. I avsnittet 2.1 nämndes energinivån för termisk strålning och atomens elektronskal som exempel på sådana diskreta tillstånd. Andra möjligheter är att utnyttja elektronernas eller atomkärnornas rotation, som kan vara i olika diskreta positioner. Man kan till och med skapa ”konstgjorda elektronskal”, med hjälp av Josephsonövergångar. Bland annat IBM gör forskning med denna typ av lagringsutrymme för kvantbitar [10].

En Josephsonövergång kan skapas genom att två aluminiumtrådar på nanonivå får oxidera, och sedan placeras mot varandra. Dessa kyls ner tills de blir



Figur 1: Exempel på kvantbit med Joephsonövergång

supraledande. Oxiden bildar en isolator, som hindrar strömmen från att flyta fritt igenom. Trots att oxiden inte leder, kommer ändå en del ström igenom, tack vare tunneleffekten, som är ett fenomen inom kvantfysiken.

I en supraledare uppstår något som kallas för Cooper-par[11], det vill säga elektronpar som är sammanbundna trots att elektroner normalt repellerar varandra. Dessa par kan passera tack vare tunneleffekten genom Josephsonövergången[12] i figur 1, så att innanför det inrutade området kommer det att finnas ett över- eller underskott av sådana par som laddar kondensatorerna. I kretsen finns två kondensatorer: kondensatorn C och Josephsonövergången, som också har sin kapacitans. Deras laddning beror på Josephsonövergången[13]. De olika laddnings nivåerna man kan mäta upp står för de olika tillstånden. Med hjälp av justering av spänningen U kan man påverka laddningen och till och med skapa superpositioner[14].

Utöver dessa finns en uppsjö av kvantfenomen som kan användas för att skapa ett lagringsutrymme för kvantbitar. En stor del av forskningen går just ut på att hitta ett stabilt och lätthanterat lagringsutrymme för kvantbitar.

Oberoende av underliggande fenomen, väljs två tillstånd och man kallar det ena tillståndet för $|0\rangle$ och det andra tillståndet för $|1\rangle$[1]. Till skillnad från den deterministiska klassiska datorn, så är det en viss sannolikhet att kvantbiten finns i det ena eller det andra tillståndet. När man inte vet i vilket tillstånd biten är i, säger man att den är i en superposition.

[1]Beteckningen förklaras närmare då den matematiska representationen förklaras.

2.3 Schrödingers katt och superposition

Få tankeexperiment har fått en sådan spridning som ”Schrödingers katt”, kanske just på grund av hur groteskt och på samma gång motsägelsefullt det är. Tankeexperiment beskrevs av Erwin Schrödinger 1935, som en kritik till Köpenhamnstolkningen av kvanttillstånd [15].

Tankeexperimentet kan beskrivas på följande sätt:

> Anta att du placerar en katt i en metallåda. I lådan finns också en Geigermätare tillsammans med en liten mängd av ett radioaktivt ämne. Geigermätaren är kopplad till en hammare och under hammaren finns en flaska med cyanid. Sannolikheten för att det radioaktiva ämnet sönderfaller under en timme är lika stor som att inget händer. Om ämnet sönderfaller, slår hammaren sönder flaskan med giftet, varvid katten dör. Denna anordning skyddas på ett sådant sätt att katten, som stängdes in i lådan, inte kan påverka situationen.

I detta tankeexperiment kopplas en kvantmekanisk händelse (sönderfallet av en atom) till en händelse på makronivå (kattens liv). Enligt den Köpenhamnska tolkningen är katten både levande och död samtidigt, tills man observerar katten. Schrödinger ville med tankeexperimentet visa hur absurda situationer denna tolkning ger och att det finns brister i förståelsen av kvantmekaniska fenomen.

Trots Schrödingers egentliga avsikt ger detta tankeexperiment en bild av vad en superposition är. När en kvantbit är i en superposition, betyder det att den har en viss sannolikhet att vara antingen $|0\rangle$ eller $|1\rangle$ då den observeras, precis som katten har en viss sannolikhet att observeras som antingen död eller levande då lådan öppnas. När en kvantbit observeras kollapsar den till någondera tillståndet.

2.4 Sammanflätade kvantbitar

Inom kvantmekaniken finns ett fenomen som ledde till en lång argumentation mellan Einstein och Bohr, nämligen sammanflätade partiklar. Einstein argumenterade för att sammanflätningen bröt mot relativitetsteorin och att förståelsen av kvantmekaniken kan inte vara korrekt[16]. Experiment från 1976 framåt verkar visa att Bohr hade rätt[17].

Sammanflätade partiklar bildar ett system tillsammans. Om man har två partiklar i ett sammanflätat tillstånd och man mäter den ena, så kan man genast säga i vilket tillstånd den andra är i, nämligen det motsatta tillståndet. Just detta ansåg Einstein vara fel, då informationen om att ena partikeln har blivit observerad inte kan nå den andra snabbare än ljuset.

Det bör dock noteras att en sammanflätning inte kan användas för att skicka information, då sammanflätningen upphör vid en observation[18]. Man kan alltså inte manipulera en kvantbit efter en mätning och förvänta sig att den motsatta manipulationen sker hos den andra kvantbiten trots att de var sammanflätade innan mätningen.

2.5 Matematisk representation

2.5.1 Kvantbitar är vektorer

Till skillnad från den klassiska datorn, där man har endast 0 och 1, hanterar man sannolikheter i en kvantdator. Varje superposition har ju en viss sannolikhet för $|0\rangle$ och $|1\rangle$. För att beskriva kvantbitarnas tillstånd matematiskt, ser man dem som vektorer i ett vektorrum.

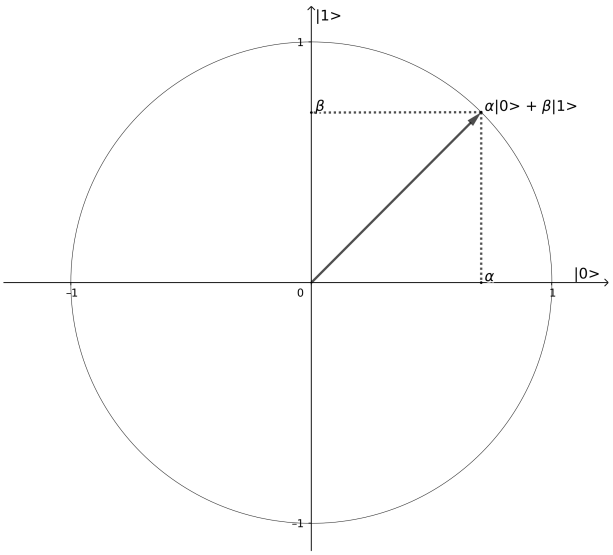
Ett vektorrum är en samling vektorer, som kan adderas med varandra och också multipliceras med skalärer. Alla dessa skalärer tillhör en bestämd mängd, som kan vara heltal, rationella tal, reella tal eller till och med komplexa tal, som i fallet med kvantbitar. Dessutom finns det i vektorrummet basvektorer och varje vektor i vektorrummet kan bildas av dessa[19]. En normaliserad vektor har alltid längden 1.

För att få en bättre bild av vad detta innebär så kan man ta det välkända koordinatsystemet med axlarna x och y. En vektor i detta koordinatsystem startar från $(0, 0)$ och slutar i en punkt, som vi kan kalla för (α, β). Basvektorer i detta koordinatsystem är x och y och alla vektorer kan skrivas som $\alpha x + \beta y$.

Då en kvantbit beskrivs som en vektor har man basvektorerna $|0\rangle$ och $|1\rangle$. En kvantbit kan då beskrivas som en normaliserad vektor $\alpha\,|0\rangle + \beta\,|1\rangle$, där α och β kan vara komplexa tal[20]. Detta är i förenklad form illustrerat i figur 2[2].

I figur 2 ser man att α och β kan bestämmas med hjälp av Pythagoras sats och

[2]Bloch-klotet ger en bättre bild, men den enkla tvådimensionella illustrationen är lättare att förstå och mera lämplig för denna avhandling.



Figur 2: En kvantbit är en normaliserad vektor

med hjälp av trigonometriska funktioner. Vektorn bildar hypotenusan med längden 1 i en triangel där α och β är kateter. Ifall vi har tillståndet där både $|0\rangle$ och $|1\rangle$ har lika stor sannolikhet, så blir vektorn $\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$ vilket naturligtvis också kan skrivas som $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$.

2.5.2 Bra-ket-notationen / Diracnotationen

Både $|0\rangle$ och $|1\rangle$ har redan nämnts i flera av de tidigare avsnitten. De är en del av bra-ket-notationen. Namnet bra-ket härstammar från det engelska ordet bracket, som betyder klammer. Denna notation skapades av Paul Dirac, vilket är orsaken till att notationen också kallas för Diracnotation[21].

Fastän alla tillstånd hittills har enbart berört en enda kvantbit, bör det noteras att ett tillstånd kan innefatta flera kvantbitar, till exempel $|10011001\rangle$.

Bra-ket-notationen har två delar: $\langle\phi|$ som kallas för ”bra” och $|\psi\rangle$ som kallas för ”ket”. Bra-delen är det hermiteska konjugatet av ett tillstånd. När bra och ket sätts ihop fås den inre produkten, som betecknas med $\langle\phi|\psi\rangle$ [21]. I avsnitt 2.5.3 kommer det hermiteska konjugatet att förklaras och en beskrivning på vad den inre produkten är hur den kan användas ges.

2.5.3 Matrisnotation

Förutom bra-ket-notationen kan man också skriva vektorer i matrisform[22]. Denna form är praktisk då ett kvanttillstånd ska förändras genom en kvantgrind.

Omskrivningen sker enligt följande:

$$\alpha\left|0\right\rangle + \beta\left|1\right\rangle = \begin{bmatrix}\alpha\\ \beta\end{bmatrix}.$$

$\left|0\right\rangle$ är samma sak som $1\left|0\right\rangle + 0\left|1\right\rangle)$ och $\left|1\right\rangle$ är samma sak som $0\left|0\right\rangle + 1\left|1\right\rangle$. Man kan då skriva om dessa fall som

$$\left|0\right\rangle = \begin{bmatrix}1\\ 0\end{bmatrix}, \left|1\right\rangle = \begin{bmatrix}0\\ 1\end{bmatrix}.$$

Ifall man har flera kvantbitar, kan man skriva deras gemensamma tillstånd som en tensorprodukt[22]. Detta är behövligt om man vill använda matriser för att räkna ut effekten av en kvantgrind med två (eller flera) ingångar. Tensorprodukten för endimensionella matriser (kvantbitar) fås enligt följande [23]:

$$\begin{bmatrix}a_1\\ a_2\\ \vdots\\ a_n\end{bmatrix} \otimes \begin{bmatrix}b_1\\ b_2\\ \vdots\\ b_n\end{bmatrix} = \begin{bmatrix}a_1b_1\\ a_1b_2\\ \vdots\\ a_1b_n\\ a_2b_1\\ a_2b_2\\ \vdots\\ a_2b_n\\ \vdots\\ a_nb_1\\ a_nb_2\\ \vdots\\ a_nb_n\end{bmatrix}.$$

Enligt detta kan man skriva om $\left|00\right\rangle$, $\left|01\right\rangle$, $\left|10\right\rangle$ och $\left|11\right\rangle$ som:

$$|00\rangle = \begin{bmatrix}1\\0\end{bmatrix} \otimes \begin{bmatrix}1\\0\end{bmatrix} = \begin{bmatrix}1\\0\\0\\0\end{bmatrix}, |01\rangle = \begin{bmatrix}1\\0\end{bmatrix} \otimes \begin{bmatrix}0\\1\end{bmatrix} = \begin{bmatrix}0\\1\\0\\0\end{bmatrix},$$

$$|10\rangle = \begin{bmatrix}0\\1\end{bmatrix} \otimes \begin{bmatrix}1\\0\end{bmatrix} = \begin{bmatrix}0\\0\\1\\0\end{bmatrix}, |11\rangle = \begin{bmatrix}0\\1\end{bmatrix} \otimes \begin{bmatrix}0\\1\end{bmatrix} = \begin{bmatrix}0\\0\\0\\1\end{bmatrix}.$$

Tensorprodukten är associativ och flera än två kvantbitar kan kombineras[23]:

$$\begin{bmatrix}a\\b\end{bmatrix} \otimes \begin{bmatrix}c\\d\end{bmatrix} \otimes \begin{bmatrix}e\\f\end{bmatrix} = \Big(\begin{bmatrix}a\\b\end{bmatrix} \otimes \begin{bmatrix}c\\d\end{bmatrix} \Big) \otimes \begin{bmatrix}e\\f\end{bmatrix} = \begin{bmatrix}a\\b\end{bmatrix} \otimes \Big(\begin{bmatrix}c\\d\end{bmatrix} \otimes \begin{bmatrix}e\\f\end{bmatrix} \Big).$$

Det hermiteska konjugatet $\langle\psi|$ av $|\psi\rangle$, som nämndes i avsnitt 2.5.2, kan skrivas som $\langle\psi| = [|\psi\rangle]^\dagger$. Detta fås genom att man transponerar $|\psi\rangle$, och sedan komplexkonjugerar talen[24]. Om

$$|\psi\rangle = \begin{bmatrix}a_1\\\vdots\\a_n\end{bmatrix}$$

så är

$$[|\psi\rangle]^\dagger = \begin{bmatrix}\bar{a_1} & \dots & \bar{a_n}\end{bmatrix}, a, \bar{a} \in \mathbb{C}.$$

Komplexkonjugeringen innebär att de imaginära delarna av talen byter tecken. Om $a_n = 3 + 2i$, så blir det nya värdet $\bar{a_n} = 3 - 2i$.

Den inre produkten, som också nämndes i avsnitt 2.5.2, är användbar då man vill fastställa sannolikheten att kvantbitarna är i ett visst tillstånd. Helt allmänt är alla kvanttillstånd beskrivna av vågfunktioner. Den inre produkten $\langle\psi|\psi\rangle$ kan definieras som integralen av $\psi^\dagger\psi$ [25].

Om man har ett tillstånd $\psi = \alpha\,|0\rangle + \beta\,|1\rangle$ kan man räkna ut $\langle\psi|\psi\rangle$ enligt

följande och få sannolikheten för $|0\rangle$ och $|1\rangle$:

$$\langle\psi|\psi\rangle = \begin{bmatrix}\bar{\alpha} & \bar{\beta}\end{bmatrix} \cdot \begin{bmatrix}\alpha \\ \beta\end{bmatrix} = \begin{bmatrix}\bar{\alpha}\alpha \\ \bar{\beta}\beta\end{bmatrix}.$$

Anta att figur 2 är en illustration av vektorn $\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$. I matrisform kan man skriva om denna vektor som:

$$\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle = \begin{bmatrix}\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}}\end{bmatrix}.$$

Man får sannolikheten för $|0\rangle$ och $|1\rangle$ genom att räkna ut den inre produkten:

$$\begin{bmatrix}\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}}\end{bmatrix} \cdot \begin{bmatrix}\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}}\end{bmatrix} = \begin{bmatrix}\frac{1}{2} \\ \frac{1}{2}\end{bmatrix}.$$

Alltså har både $|0\rangle$ och $|1\rangle$ sannolikheten $\frac{1}{2}$.

3 Kvantgrindar

Kvantgrindarna är byggstenarna i en kvantalgoritm. Det är dessa block som omvandlar ett kvanttillstånd till ett annat.

I avsnitt 3.1 presenteras kvantgrindar som unitära matriser. Definitionen på unitära matriser ges i samband med detta. Avsnittet visar också hur man kan matematiskt räkna ut resultatet av att en kvantgrind har opererat på ett tillstånd av kvantbitar.

Därefter presenteras tre kvantgrindar i avsnitten 3.2-3.4: nämligen NOT-, Hadamard- och CNOT-grinden. Dessa tre grindar är viktiga för att man ska förstå Deutsch-Josza-algoritmen i avsnitt 4.

3.1 Kvantgrindar som matriser

Liksom kvantbitar kan skrivas i matrisform, så har varje kvantgrind en motsvarande matris. I avsnitt 2.5.3 demonstrerades matrismultiplikation och tensorprodukt

för endimensionella matriser. För att kunna hantera kvantgrindar behövs samma operationer, men för tvådimensionella matriser.

Multiplikation av tvådimensionella matriser görs enligt följande[26]:

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \cdot \begin{bmatrix} e & f \\ g & h \end{bmatrix} = \begin{bmatrix} ae + bg & af + bh \\ ce + dg & cf + dh \end{bmatrix}.$$

Den första matrisen måste alltså ha lika många kolumner som den andra matrisen har rader.

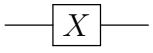
Anta att matrisen A är en kvantgrind och $|\psi\rangle$ är ett tillstånd som kvantgrinden opererar på. Då erhåller man resultatet genom att utföra matrismultiplikationen $A(|\psi\rangle)$ [27]. De följande avsnitten kommer att demonstrera detta.

Om man har två kvantgrindar, beskrivna av matriserna A och B, och man har två kvantbitar och man vill utföra A på den första kvantbiten och B på den andra kvantbiten, kan man kombinera A och B med hjälp av tensorprodukten till en enda grind som tar in två kvantbitar[22]. Tensorprodukten erhålls enligt följande:

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \otimes \begin{bmatrix} e & f \\ g & h \end{bmatrix} = \begin{bmatrix} ae & af & be & bf \\ ag & ah & bg & bh \\ ce & cf & de & df \\ cg & ch & dg & dh \end{bmatrix}.$$

Alla kvantgrindar är unitära matriser [27]. Detta betyder att om matrisen A representerar en kvantgrind, så är $A^\dagger A = AA^\dagger = I$, där I är enhetsmatrisen [28]. En enhetsmatris är en matris med enbart nollor, förutom längs diagonalen där det enbart finns ettor, vilket betyder att $AI = A$. På klarspråk betyder allt detta att till skillnad från klassiska grindar i digitalelektronik, så är kvantgrindar sådana att man kan, utgående från resultatet, räkna ut de ursprungliga kvantbitarnas tillstånd.

3.2 NOT-grinden



NOT-grinden är en grind som opererar på en enda kvantbit och resultatet är motsatsen till vad som kom in: $|0\rangle \mapsto |1\rangle$ och $|1\rangle \mapsto |0\rangle$. Detta betyder att om

kvantbiten inte är i en superposition, så beter sig NOT-kvantgrinden exakt som NOT-grinden i digitalelektronik. I det allmänna fallet, där också en superposition är möjlig, sker följande: $\alpha\,|0\rangle + \beta\,|1\rangle \mapsto \alpha\,|1\rangle + \beta\,|0\rangle$. NOT-grindens representation i matrisform är[27]

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

Hela operationen i matrisform blir:

$$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} 0 \cdot \alpha + 1 \cdot \beta \\ 1 \cdot \alpha + 0 \cdot \beta \end{bmatrix} = \begin{bmatrix} \beta \\ \alpha \end{bmatrix}.$$

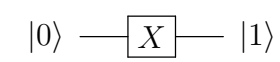
Om man har två NOT-grindar efter varandra återställs kvantbiten eftersom enhetsmatrisen fås genom:

$$XX = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = I.$$

Eftersom X är symmetrisk runt diagonalen är $X = X^\dagger$. Detta betyder att multiplikationen ovan också visar att NOT-grinden är unitär.

Exempel

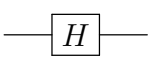
Eftersom detta är den första kvantgrinden som behandlas, är det skäl till att ta ett exempel, så att det tydligt framkommer hur man utför beräkningarna.



Om NOT-grinden opererar på $|0\rangle$ får man följande uttryck:

$$X(|0\rangle) = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \cdot 1 + 1 \cdot 0 \\ 1 \cdot 1 + 0 \cdot 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

3.3 Hadamard



Liksom NOT-grinden verkar Hadamard-grinden på en kvantbit. Hadamard-grinden används för att skapa superpositioner[27]:

$$\begin{aligned} |0\rangle &\mapsto \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle \\ |1\rangle &\mapsto \frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle \end{aligned}.$$

Detta betyder att om Hadamard-grinden opererar på en kvantbit så kommer den att vara i en superposition, där både $|0\rangle$ och $|1\rangle$ är lika sannolika. Hadamard representeras av matrisen[27]:

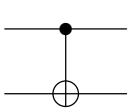
$$H = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}.$$

Om man har två Hadamard-grindar efter varandra återställs kvantbiten eftersom enhetsmatrisen fås genom:

$$\begin{aligned} HH &= \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \cdot \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \\ &= \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix} \cdot \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix} \\ &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = I. \end{aligned}$$

H är symmetrisk runt diagonalen, vilket betyder att $H = H^\dagger$. Multiplikationen ovan visar också att Hadamard-grinden är unitär.

3.4 CNOT



CNOT (Controlled NOT) opererar på två kvantbitar och representeras av matri-

sen[27]:

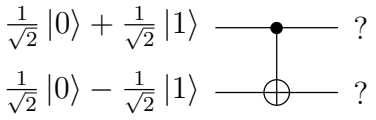
$$CNOT = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}.$$

Om kvantbitarna tillhör mängden $\{|0\rangle, |1\rangle\}$, kan CNOT till en början likna XOR från digitalelektroniken, eftersom CNOT utför en NOT-operation på den andra kvantbiten (resultatkvantbiten), enbart om den första kvantbiten (kontrollkvantbiten) är $|1\rangle$. Den första kvantbiten förändras i detta fall inte av kvantporten[29]:

$$\begin{aligned} |00\rangle &\mapsto |00\rangle \\ |01\rangle &\mapsto |01\rangle \\ |10\rangle &\mapsto |01\rangle \\ |11\rangle &\mapsto |10\rangle \end{aligned}.$$

Resultatkvantbiten för CNOT-grinden blir enbart $|1\rangle$ om båda ingångarna har motsatta värden.

Om man utvidgar mängden av tillstånd en kvantbit kan ha till samtliga, märker man att CNOT-grinden ibland byter roller för kontroll- och resultatingångarna. Anta att första kvantbiten är $\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$ och andra kvantbiten är $\frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle$. Vad blir resultatet?



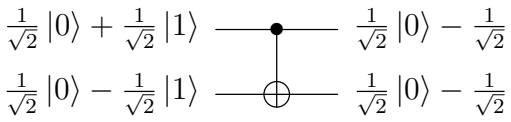
För att bestämma resultatet behöver man först räkna ut tensorprodukten av kvantbitarna:

$$\begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix} \otimes \begin{bmatrix} \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} \end{bmatrix} = \begin{bmatrix} \frac{1}{2} \\ -\frac{1}{2} \\ \frac{1}{2} \\ -\frac{1}{2} \end{bmatrix}.$$

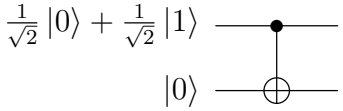
Därefter multipliceras CNOT-matrisen med tillståndet ovan:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} \frac{1}{2} \\ -\frac{1}{2} \\ \frac{1}{2} \\ -\frac{1}{2} \end{bmatrix} = \begin{bmatrix} \frac{1}{2} \\ -\frac{1}{2} \\ -\frac{1}{2} \\ \frac{1}{2} \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} \end{bmatrix} \otimes \begin{bmatrix} \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} \end{bmatrix}.$$

Den första kvantbiten har alltså ändrats och den andra passerade oförändrad. Detta är en princip som utnyttjas i Deutsch-Josza-algoritmen, som presenteras i avsnitt 4.



Vidare är CNOT-grinden användbar för att skapa sammanflätade kvantbitar[29]. Ett sådant tillstånd kan skapas om kontrollingången är $\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$ och den andra ingången är $|0\rangle$.



Tensorprodukten av ingångarna fås enligt:

$$\begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix} \otimes \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ 0 \\ \frac{1}{\sqrt{2}} \\ 0 \end{bmatrix}.$$

Därefter multipliceras CNOT-matrisen med tillståndet ovan:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} \frac{1}{\sqrt{2}} \\ 0 \\ \frac{1}{\sqrt{2}} \\ 0 \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ 0 \\ 0 \\ \frac{1}{\sqrt{2}} \end{bmatrix}.$$

Efter denna uträkning har man ett tillstånd som inte kan formas som en tensorprodukt av två enskilda kvantbitar[30]. Kvantbitarna måste beskrivas tillsammans, de

är sammanflätade: $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$.

Eftersom CNOT är symmetrisk runt diagonalen så är $CNOT = CNOT^{\dagger}$. CNOT är en unitär grind eftersom enhetsmatrisen I fås genom multiplikationen $CNOT \cdot CNOT^{\dagger}$:

$$CNOT \cdot CNOT^{\dagger} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = I$$

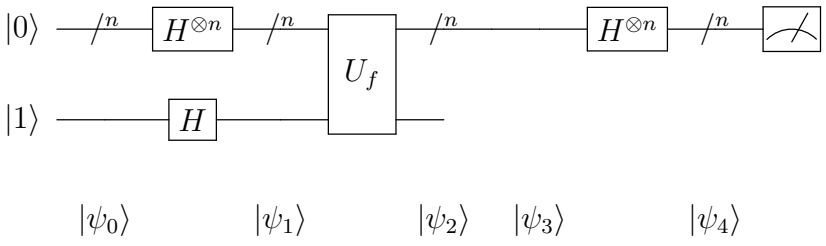
4 Deutsch-Jozsa-algoritmen

Anta att man får en krets med n ingångar och att resultatet kan avläsas från en enda utgång. Kretsen kan beskrivas som en funktion $f : \{0,1\}^n \mapsto \{0,1\}$. Man vet att denna funktion är garanterat antingen balanserad eller konstant, men inte vilkendera. Uppgiften är att ta reda på vilkendera typ av funktion kretsen har implementerat[31].

En balanserad funktion är en funktion som ger 0 för hälften av inmatningen och 1 för andra hälften. Till exempel är en funktion som ger 1 för udda tal och 0 för jämna tal en sådan funktion. En annan balanserad funktion är en som ger 0 för den lägre hälften av all indata och 1 för den högre hälften av all indata. I kvantdatorvärlden är kvantgrindarna NOT och CNOT exempel på balanserade grindar. En konstant funktion ger alltid samma svar oberoende av vad som matades in i kretsen, till exempel alltid 0 för alla inmatningar. En konstant funktion har alltså antingen formen $f(x) = 0$ eller $f(x) = 1$ för alla värden på x.

Eftersom det finns n ingångar, finns det 2^n möjligheter av indata. En klassisk dator behöver i bästa fallet testa enbart två av dem, nämligen om ena försöket ger 0 och andra försöket ger 1 kan man vara säker på att funktionen är balanserad. Som mest kan man behöva göra $2^{n-1} + 1$ tester, det vill säga en mera än hälften av alla möjligheter. Om alla dessa ger samma värde kan man vara säker på att funktionen är konstant.

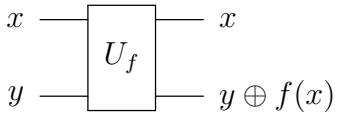
Om funktionen f kan implementeras som en kvantkrets kan Deutsch-Jozsa-algoritmen lösa detta problem med ett enda test oberoende av vilken balanserad



Figur 3: Deutsch-Jozsa-algoritmen

eller konstant funktion som implementerades. Det finns alltså inget värsta fall, som för den klassiska datorn.

Deutsch-Jozsa algoritmen är presenterad som en kvantkrets i figur 3. Grinden U_f är konstruerad så att den har $n+1$ ingångar. Hela grinden är unitär. Funktionen f, som är antingen konstant eller balanserad, körs på de första n ingångarna och resultatet är sedan kopplat till en CNOT-grind. Den sista ingången, y, är också kopplad till samma CNOT-grind. Effekten av U_f är alltså $|x\rangle\,|y\rangle \mapsto |x\rangle\,|y \oplus f(x)\rangle$.



De första n kvantbitarna, de som ska bli indata för f, sätts till $|0\rangle$ och den sista ingången, som skall bli indata för CNOT, sätts till $|1\rangle$. Alla kvantbitar går igenom Hadamard-grindar innan de når U_f. Vi har alltså $n+1$ Hadamard-operationer.

För att få veta vad tillståndet är efter n Hadamard-operationer på $|0\rangle$ måste man först ta reda på hur tensorprodukten $H^{\otimes n}$ ser ut. Får att få en bättre förståelse, kan man utföra tensorprodukten för $H^{\otimes 2}$ och $H^{\otimes 3}$.

En ensam Hadamardgrind har matrisen:

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

Två Hadamardgrindar har matrisen:

$$H \otimes H = \frac{1}{\sqrt{2^2}} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}.$$

Tre Hadamardgrindar har matrisen:

$$H \otimes H \otimes H = \frac{1}{\sqrt{2^3}} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \end{bmatrix}.$$

Eftersom varje värde i matrisen är antingen -1 eller 1 inser man lätt att för varje gång man ökar tensorprodukten med en Hadamard-matris blir skalären $\frac{1}{\sqrt{2}}$ multiplicerad med den föregående skalären. För n Hadamard-grindar blir skalären $\frac{1}{\sqrt{2^n}}$. Man ser också att hela första kolumnen i den sammanslagna Hadamard-matrisen allid kommer att vara fylld med 1, eftersom $1 \cdot 1 = 1$ och från definitionen av tensorprodukt ser man att andra operationer inte är möjliga för denna kolumn då originalmatrisen redan var fylld med 1 för denna kolumn.

Eftersom de första n kvantbitarna är alla $|0\rangle$ blir tensorprodukten:

$$|0\rangle^{\otimes n} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}.$$

Antalet rader är 2^n. Resultatet av att Hadamard-grindarna opererar på detta tillstånd

är:

$$H^{\otimes n}(|0\rangle^{\otimes n}) = \frac{1}{\sqrt{2^n}} \begin{bmatrix} 1 \\ 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}.$$

Detta skrivs om som:

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

Den sista kvantbiten har värdet $|1\rangle$ och, som tidigare konstaterat, är $H(|1\rangle) = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$. Därför blir resultatet av Hadamard-grinden på $n+1$ kvantbitar där de n första kvantbitarna är $|0\rangle$ och den sista kvantbiten $|1\rangle$:

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

Eftersom effekten av U_f är $|x\rangle \, |y\rangle \mapsto |x\rangle \, |y \oplus f(x)\rangle$ ser man att

$$U_f(|x\rangle \, |0\rangle) = |x\rangle \, |0 \oplus f(x)\rangle = |x\rangle \, |f(x)\rangle$$
$$U_f(|x\rangle \, |1\rangle) = |x\rangle \, |1 \oplus f(x)\rangle .$$

Detta leder till att man kan omskriva $U_f(|x\rangle \, (|0\rangle - |1\rangle))$ som $|x\rangle \, (|f(x)\rangle - |1 \oplus f(x)\rangle)$ Om U_f får operera på $|\psi_1\rangle$ fås:

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

Funktionen $f(x)$ kan enbart anta värdet 0 eller 1. Om man sätter in $f(x) = 0$

respektive $f(x) = 1$ får man

$$|\psi_2\rangle = \frac{1}{\sqrt{2^{n+1}}} \sum_{x=0}^{2^{n-1}} |x\rangle \left(|0\rangle - |1\rangle\right) \text{ om } f(x) = 0 \text{ och}$$
$$|\psi_2\rangle = \frac{1}{\sqrt{2^{n+1}}} \sum_{x=0}^{2^{n-1}} |x\rangle \left(|1\rangle - |0\rangle\right) \text{ om } f(x) = 1.$$

Det sker alltså ett teckenbyte om $f(x) = 1$. Man kan använda sig av $(-1)^{f(x)}$ för detta teckenbyte:

$$|\psi_2\rangle = \frac{1}{\sqrt{2^{n+1}}} \sum_{x=0}^{2^{n-1}} (-1)^{f(x)} |x\rangle \left(|0\rangle - |1\rangle\right).$$

Från Figur 3 ser man att från och med nu framåt behövs inte mera den sista kvantbiten. Tillståndet man behöver beakta är

$$|\psi_3\rangle = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^{n-1}} (-1)^{f(x)} |x\rangle .$$

Eftersom man enligt algoritmen ska göra en Hadamard-transformation på detta, behövs mera kunskap om tensorprodukten av Hadamard-matriser. Hadamard-matrisen kan skrivas om som[32]:

$$H = \frac{1}{\sqrt{2}} \begin{array}{c} \\ 0 \\ 1 \end{array} \begin{array}{c} \begin{array}{cc} 0 & 1 \end{array} \\ \begin{bmatrix} (-1)^{0\wedge 0} & (-1)^{0\wedge 1} \\ (-1)^{1\wedge 0} & (-1)^{1\wedge 1} \end{bmatrix} \end{array}.$$

Man tar radnumret och kolumnnumret (som binära tal) och utför booleska AND-operationen. Resultatet av AND-operationen blir antingen 0 eller 1. Den ursprungliga Hadamard-matrisen återfås då $(-1)^0 = 1$ och $(-1)^1 = -1$.

Om man har tensorprodukten av två Hadamard-matriser får man:

$$H \otimes H = \frac{1}{\sqrt{2^2}} \begin{array}{c} \\ 00 \\ 01 \\ 10 \\ 11 \end{array} \begin{array}{c} \begin{array}{cccc} 00 & 01 & 10 & 11 \end{array} \\ \begin{bmatrix} (-1)^{0\wedge 0\oplus 0\wedge 0} & (-1)^{0\wedge 0\oplus 0\wedge 1} & (-1)^{0\wedge 1\oplus 0\wedge 0} & (-1)^{0\wedge 1\oplus 0\wedge 1} \\ (-1)^{0\wedge 0\oplus 1\wedge 0} & (-1)^{0\wedge 0\oplus 1\wedge 1} & (-1)^{0\wedge 1\oplus 1\wedge 0} & (-1)^{0\wedge 1\oplus 1\wedge 1} \\ (-1)^{1\wedge 0\oplus 0\wedge 0} & (-1)^{1\wedge 0\oplus 0\wedge 1} & (-1)^{1\wedge 1\oplus 0\wedge 0} & (-1)^{1\wedge 1\oplus 0\wedge 1} \\ (-1)^{1\wedge 0\oplus 1\wedge 0} & (-1)^{1\wedge 0\oplus 1\wedge 1} & (-1)^{1\wedge 1\oplus 1\wedge 0} & (-1)^{1\wedge 1\oplus 1\wedge 1} \end{bmatrix} \end{array}.$$

I detta sammanhang är $\oplus$ boolesk XOR. Denna metod fungerar även för att skapa $H^{\otimes n}$-matriser [33]:

$$H^{\otimes n}[x, y] = \frac{1}{\sqrt{2^n}}(-1)^{<x,y>}$$

där x och y är rad- respektive kolumnnumret som binära tal och x_n och y_n står för de enskilda bitarna i talet och $< x, y >$ är $x_1 \wedge y_1 \oplus x_2 \wedge y_2 \oplus \ldots \oplus x_n \wedge y_n$.

Då Hadamard-tranformation görs på $|\psi_3\rangle$ får man:

$$\begin{aligned} |\psi_4\rangle &= \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^{n-1}} (-1)^{f(x)} |x\rangle \left[\frac{1}{\sqrt{2^n}} \sum_{y=0}^{2^{n-1}} (-1)^{<x,y>} |y\rangle \right] \\ &= \frac{1}{2^n} \sum_{x=0}^{2^{n-1}} \left[\sum_{y=0}^{2^{n-1}} (-1)^{f(x)} (-1)^{<x,y>} \right] |y\rangle \\ &= \frac{1}{2^n} \sum_{x=0}^{2^{n-1}} \left[\sum_{y=0}^{2^{n-1}} (-1)^{f(x)\oplus <x,y>} \right] |y\rangle . \end{aligned}$$

Om man vill bestämma sannolikheten för att uttrycket kollapsar till $|0\rangle$, behöver y vara 0 [31]. Om man beaktar att $< x, y >=< x, 0 >= 0$ för alla värden på x får man uttrycket reducerat till[32]:

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

Här ser man att $|\psi_4\rangle$ helt beror på $f(x)$. Eftersom $(-1)^{f(x)}$ är ett reellt tal kan man direkt kvadrera utan att komplexkonjugerara för att få sannolikheten för $|0\rangle$.

Om $f(x)$ är konstant 0 för alla värden på x är sannolikheten för att resultatet är $|0\rangle$:

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

Om $f(n)$ är konstant 1 för alla värden på x är sannolikheten för att resultatet är $|0\rangle$:

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

Om $f(n)$ är balanserad kommer hälften av kvantbitarna att ta ut den andra hälften och sannolikheten för att resultatet är $|0\rangle$:

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

Man får alltså ett deterministiskt svar på första försöket: $|0\rangle$ om funktionen är konstant och $|1\rangle$ om funktionen är balanserad.

5 Sammanfattning och slutsats

Denna avhandling visade att kvantdatorer kan lösa en del problem snabbare än vad en klassisk dator kan lösa samma problem. Detta visades genom att demonstrera Deutsch-Joza-algoritmen, som löser ett problem där man avgör om en funktion är balanserad eller konstant med ett enda försök. En klassisk kan behöva upp till $2^n + 1$ försök för att lösa samma problem.

För att kunna visa detta, presenterades några exempel på kvantmekaniska fenomen så att man kan få en idé om vad en kvantbit är och hur den kan implementeras. Det handlar om diskreta kvantfenomen som väljs ut att representera $|0\rangle$ och $|1\rangle$. Kvantbitens tillstånd beskrivs som en vektor $\alpha |0\rangle + \beta |1\rangle$. En kvantbit kan vara antingen $|0\rangle$ eller $|1\rangle$ eller i en superposition där $|0\rangle$ och $|1\rangle$ har vardera en viss sannolikhet.

Vidare presenterades också sammanflätade kvantbitar, eftersom detta är ett

fenomen som kan uppstå med de kvantgrindar som behandlades. Sammanflätade kvantbitar kan enbart beskrivas tillsammans som ett system och då ena kvantbiten avläses kommer den andra att vara i motsatt position. Ett exempel med sammanflätade kvantbitar presenterades tillsammans med kvantgrinden CNOT.

Kvantgrindarna NOT, Hadmard, och CNOT presenterades eftersom de behövs för att man ska förstå Deutsch-Jozsa algoritmen. Alla kvantgrindar är unitära, det vill säga man kan från resultatet av en grind räkna ut vad indatan var genom att använda hermiteska konjugatet av grinden. Detta skiljer sig från digitalelektroniken där de flesta grindarna förstör data. Till exempel XOR, som påminner om CNOT har två ingångar men enbart en utgång, medan CNOT har två ingångar och två utgångar.

Även U_f, grinden som implementerar funktionen f, som testas i Deutsch-Joszaalgoritmen, behandlades. Denna grind implementerar f, som är garanterad att vara antingen balanserad eller konstant, och kopplar resultatet av f till en inbyggd CNOT-grind. U_f består alltså av både f och CNOT och är som en helhet en unitär operator.

Denna avhandling har gett en grund som hjälper en att bättre förstå vad kvantdatorer handlar om, utan att kräva djupa förkunskaper i matematik och fysik. Med de grundprinciper, som avhandlingen har behandlat är det lättare att ta del av nyhetsflödet och att också studera vidare om kvantdatorer.

Förutom universitet bedriver många stora företag, som IBM, Google, Microsoft och D-Wave Systems aktiv forskning inom området. Helt säkert kommer nya arbetsplatser att skapas inom denna bransch, och det borde motivera till att studera ämnet djupare.

Referenser

[1] Max Riedel m. fl. “The European quantum technologies flagship programme”. I: *Quantum Science and Technology* 2.3 (2017).

[2] Max Planck. *The Theory of Heat Radiation*. 1914.

[3] *Svartkropp*. 2018. URL: https://sv.wikipedia.org/wiki/Svartkropp.

[4] *Energy of Photon using E= nhv*. 2018. URL: http://yeahchemistry.com/tutorials/energy-photon-using-e-nhv.

[5] Niels Bohr. “On the Constitution of Atoms and Molecules”. I: *Philosophical Magazine* (1913).

[6] *Kvantmekanik*. 2018. URL: https://sv.wikipedia.org/wiki/Kvantmekanik.

[7] David Finkelstein. “Space-Time Structure in High Energy Interactions”. I: *Gudehus* (1968).

[8] Jurij Manin. *Vychislimoe i nevychislimoe (Beräkningsbart och icke beräkningsbart*. 1980.

[9] *Kvantdator*. 2018. URL: https://sv.wikipedia.org/wiki/Kvantdator.

[10] Christine Vu. *IBM Makes Quantum Computing Available on IBM Cloud to Accelerate Innovation*. 2016. URL: https://www-03.ibm.com/press/us/en/pressrelease/49661.wss.

[11] *Cooper pair*. 2018. URL: https://en.wikipedia.org/wiki/Cooper_pair.

[12] *Josephson effect*. 2018. URL: https://en.wikipedia.org/wiki/Josephson_effect.

[13] *Superconducting quantum computing*. 2018. URL: https://en.wikipedia.org/wiki/Superconducting_quantum_computing.

[14] *Charge qubit*. 2018. URL: https://en.wikipedia.org/wiki/Charge_qubit.

[15] Erwin Schrödinger. “Die gegenwärtige Situation in der Quantenmechanik”. I: *Naturwissenschaften* (48 1935).

[16] Albert Einstein, Boris Podolsky och Nathan Rosen. “Can Quantum-Mechanical Description of Physical Reality be Considered Complete?” I: *Physical Review* (1935).

[17] David Bates. *Advances in atomic and molecular physics*.

[18] *Kvantsammanflätning*. 2018. URL: https://sv.wikipedia.org/wiki/Kvantsammanfl%C3%A4tning.

[19] *Vector space*. 2018. URL: https://en.wikipedia.org/wiki/Vector_space.

[20] *Qubit*. 2018. URL: https://en.wikipedia.org/wiki/Qubit.

[21] *Bra–ket notation*. 2018. URL: https://en.wikipedia.org/wiki/Bra%E2%80%93ket_notation.

[22] *What is Quantum Computing?* Microsoft. 2017. URL: https://docs.microsoft.com/en-us/quantum/quantum-concepts-1-intro?view=qsharp-preview.

[23] *Tensor product*. 2018. URL: https://en.wikipedia.org/wiki/Tensor_product.

[24] *Hermiteskt konjugat*. 2018. URL: https://sv.wikipedia.org/wiki/Hermiteskt_konjugat.

[25] *Wave function*. 2018. URL: https://en.wikipedia.org/wiki/Wave_function.

[26] *Matrix multiplication*. 2018. URL: https://en.wikipedia.org/wiki/Wave_function.

[27] *Quantum logic gate*. 2018. URL: https://en.wikipedia.org/wiki/Quantum_logic_gate.

[28] *Unitary matrix*. 2018. URL: https://en.wikipedia.org/wiki/Unitary_matrix.

[29] *Controlled NOT gates*. 2018. URL: https://en.wikipedia.org/wiki/Controlled_NOT_gate.

[30] Utkarsh Sinha. *Quantum Computing Explained*. URL: https://www.clerro.com/guide/580/quantum-computing-explained.

[31] *Deutsch–Jozsa algorithm*. 2018. URL: https://en.wikipedia.org/wiki/Deutsch%E2%80%93Jozsa_algorithm.

[32] Noson S. Yanofsky. *Quantum Computing for Computer Scientists*. Cambridge University Press, 2013.

[33] *Hadamard transform*. 2018. URL: https://en.wikipedia.org/wiki/Hadamard_transform.