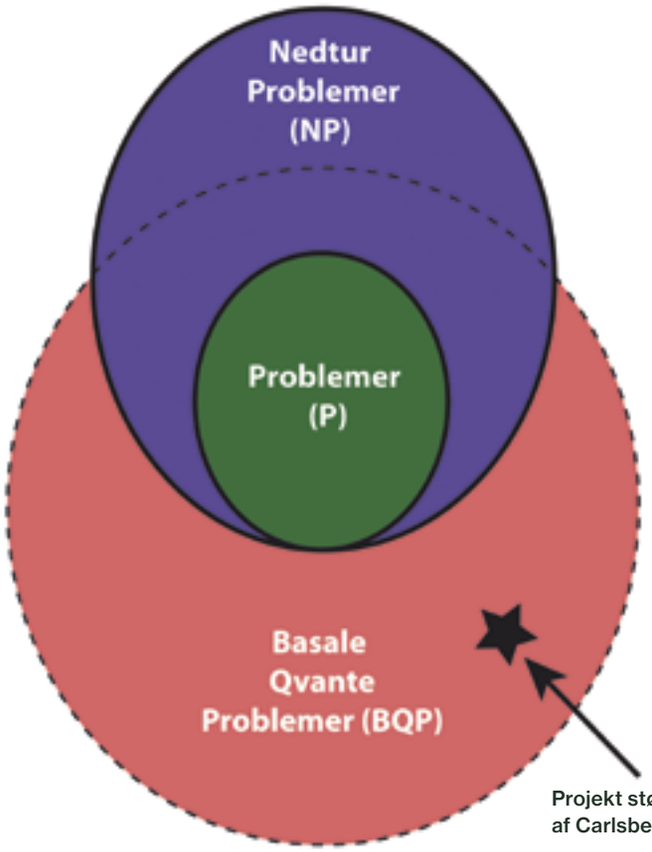
8

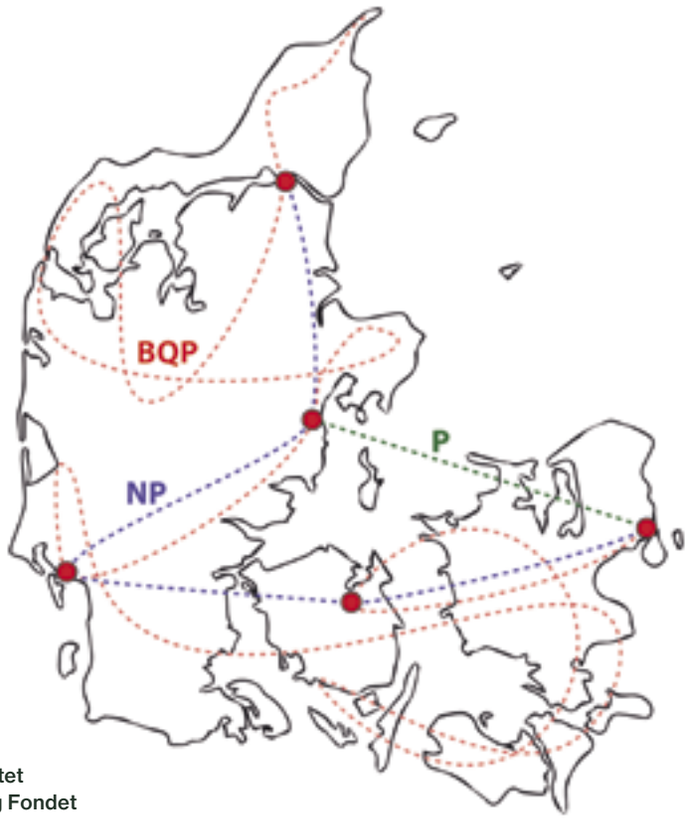
NYE TERRITORIER PÅ PROBLEM-ERNES VERDENSKORT

Af
MORTEN KJAERGAARD
PH.D., POSTDOCSTIPENDIAT VED MASSACHUSETTS INSTITUTE OF TECHNOLOGY

MODTAGET ET CARLSBERG FOUNDATION INTERNATIONALISATION FELLOWSHIP TIL PROJEKTET *FAULT TOLERANCE AND QUANTUM ALGORITHMS IN 3D INTEGRATED SUPERCONDUCTING ERROR CORRECTED QUANTUM PROCESSORS*

Selvom almindelige computere bliver hurtigere og mindre hele tiden, har de stadig nogle ultimative begrænsninger. En kvantecomputer er ikke underlagt de samme begrænsninger og kan måske ændre fundamentalt på hvad vi bruger computere til i fremtiden. Det viser sig at være vanvittigt krævende at konstruere kvantecomputere og et internationalt kapløb med både universiteter, tech-giganter og start-up firmaer er i fuld gang for at bygge og udnytte kvantecomputerens unikke kræfter. Morten Kjaergaards forskningsprojekt 'Fault tolerance and quantum algorithms in 3D integrated superconducting error corrected quantum processors' er en del af det kapløb.





Projekt støttet
af Carlsberg Fondet

P eksempel:
Hvad er den korteste vej mellem København og Aarhus?

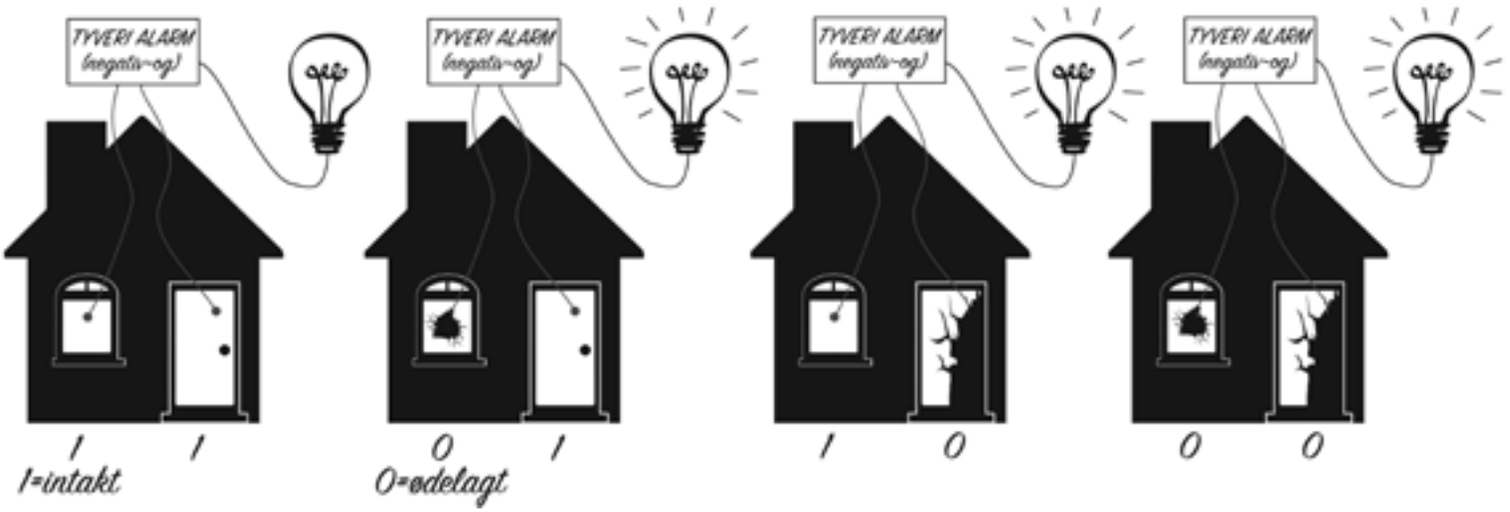
NP eksempel:
Er der en rute på mindre end 700 kilometer som forbinder Aalborg, Aarhus, Odense, Esbjerg og København?

BQP eksempel:
Er der en knude på Danmarksruten vist ovenfor?

Måske har du ikke tænkt over det, men computerkraft er overalt omkring os – hver gang du går på nettet, hver gang du bruger en GPS eller ringer med din telefon, bruger du i dag computerkraft. Og selvom både internetsøgninger, vejvisning med GPS og telefonopkald allesammen føles som om de går hurtigt, er der stadig et væld af tilfælde i verden hvor mere computerkraft ville gavne. Eksempelvis bruges gigantiske computere flittigt til at lave digitale tests af nye vaccinekandidater inden man starter dyre og krævende eksperimenter på levende dyr. Store udregninger på computere bruges også til at finde den optimale form på vingen af en vindmølle så man får maximal udnyttelse af vindenergi, og til at finde den sejlrute der kræver mindst brændstof for store tankskibe. Kort sagt: Computerkraft spiller en central rolle i vores verden – lidt ligesom kulkraft i det forrige århundrede, selvom man ikke altid ser den lige så tydeligt som osende kulkraftværker.

Men selvom computere bliver hurtigere og hurtigere, ved vi også at over for visse udfordringer vil selv de absolut største supercomputere altid ende med at give op. Kvantecomputeren derimod kan have mere muskelkraft end sådanne klassiske supercomputere, og nogle af de problemer som supercomputerne i dag giver op over for, kan have store samfundsmæssige konsekvenser hvis bare vi kunne løse dem. I mit forskningsprojekt på Massachusetts Institute of Technology i Boston har jeg arbejdet på udvikling af kraftfulde programmer til fremtidens kvantecomputere og metoder til at gøre dem mere robuste.

Figur 1
Et forsimplet verdenskort over problemer og eksempler fra hver problemkategori. Bemærk overlappet mellem BQP og NP: Visse NP-problemer kan ikke løses effektivt på en klassisk supercomputer, men er nemme at løse på en kvantecomputer. Det betyder at kvantecomputere har åbnet op for en helt ny kategori af problemer, vi kan løse. Det er lige nu et meget aktivt forskningsfelt at finde ud af præcist hvilke problemer der hører til hvor, og i sit projekt har Morten Kjaergaard arbejdet på et af disse problemer (markeret med en stjerne).



Figur 2
En tyverialarm bruger et såkaldt 'negativ-og' logisk skridt. Hvis vinduet og døren (eller begge dele) er ødelagt, lyser lampen og indikerer, at der er indbrud. Hvis begge dele er intakte er lyset slukket.

Et landkort over problemernes verden

Alan Turing (1912-1954) levede i England. Han var matematiker og den første som tænkte abstrakt over hvad en computer egentlig er. Han opfandt stort set egenhændigt det vi i dag kalder teoretisk computervidenskab i 1937[1] (efter han havde gjort dette, hjalp han i øvrigt med at knække nazisternes krypterede kommunikationssystemer under 2. verdenskrig og spillede dermed en stor rolle i at de allierede vandt krigen!). Det er ingen overdrivelse at sige at hvad Platon gjorde for filosofien, gjorde Alan Turing for computervidenskab. Turings ideer bidrog til at vi nu kan svare helt konkret på hvad der er svært, og hvad der er nemt for vores computere at regne ud. Selvom det måske lyder lidt tørt, så viser det sig at have dramatiske konsekvenser.

På grund af Turings indsigt kan vi nu svare konkret på meget fundamentelle spørgsmål om matematik, tal og ligninger: Er det rent faktisk sværere at gange end at plusse? Det var det i hvert fald i folkeskolen, men er det i al almindelighed rigtigt, eller var vi bare lidt for dovne til at genkende mønstret dengang? Nu er gange og plusse måske ikke de mest interessante ting man kan gøre på en computer, men man kunne også være interesseret i at spørge: Kan jeg udvikle en generel computermodel af en vaccine og bruge den til at forudsige om vaccinen virker uden nogensinde at udføre forsøg med mennesker? Den slags spørgsmål vil vi meget gerne have svar på, og Turings modeller kan bruges som en rettesnor for at finde ud af om computere overhovedet kan bruges til at besvare sådanne spørgsmål.

Problem-verdenskortet

Uden at vide det benytter du stort set hver dag teknologi baseret på at NP-problem-landet er større end P-landet. Et eksempel: Når du logger ind på Netbank, går du ud fra at en internet-tyv ikke kan knække din adgangskode. Det kan du også føle dig tryg ved fordi det viser sig at det matematiske trick som man bruger til sikre internet-trafik, er et NP-problem: svært at løse (at knække din kode), men nemt at tjekke (som din bank gør for at sikre at du er dig). Hvis det matematiske trick som netbank bruger, var et P-problem, så kunne tyven bryde din kode lige så let som banken kan tjekke at du er dig. Problem-verdenskortet har selvfølgelig rigtig mange andre lande (nogle af dem mystiske og eksotiske). Kvantecomputer-landet BQP blev først 'opdaget' i 1990'erne.[2] De lidt pudsige forkortelser stammer fra de originale tekniske betegnelser for problemkategorierne. For eksempel står 'NP' for 'Non-deterministic Polynomial time on a Turing machine' – noget af en mundfuld, så vi holder os til 'Nedtur-problemer' i stedet for.

For at finde rundt i sværhedsgraden af problemer, kan man bruge Turings ideer til at lave en slags verdenskort over hvor svære problemer er at løse. Ligesom på et normalt verdenskort er der lande og grænser mellem lande. I Figur 1 ser du et meget forsimplet 'problem-verdenskort'. Jeg har kun indtegnet de tre store problem-kontinenter (tænk på et verdenskort kun med Europa, Afrika og Asien uden landegrænser indtegnet). Kvantecomputerens kræfter kan bedst forstås ved at se, hvordan problem-verdenskortets grænser flyttes når vi kan bruge kvantekræfterne til at løse problemer med.

Grænserne imellem landene på problem-verdenskortet er bestemt af hvor svært det er at finde løsningen til et problem i det land. Et par eksempler gør det nemmere at forstå: Ikke-alt-for-svære problemer (dem vi bare kalder for 'problemer' og giver symbolet 'P') er dem hvor det er relativt nemt at finde en løsning, og det er relativt nemt at tjekke om løsningen er rigtig (f.eks. gange og plusse). Svære problemer (dem vi kalder for 'nedtur-problemer' og giver dem symbolet 'NP') er karakteriseret ved at det er hamrende svært at finde en løsning, men relativt nemt at tjekke om en løsning er korrekt. Så hvis nogen siger at de har løst et NP-problem, kan du hurtigt tjekke om det er korrekt. Selvfølgelig er det sådan at hvis man kan løse NP-problemer, så kan man også løse P-problemer, og derfor er P's landegrænse inden for NP's landegrænse. Det sidste problem-land er BQP ('Basale Qvante Problemer'), der er nemme at løse og tjekke på kvantecomputere.

Det utrolige, som har gjort at teknik-giganter som Google, IBM og Microsoft (bl.a. i København) investerer store summer i kvantecomputer-udvikling, er at nogle af de praktisk talt umulige problemer fra NP (nogle af dem har store økonomiske og måske sociale konsekvenser) er nemme at løse på en kvantecomputer! Hvad der før var umuligt, bliver altså muligt hvis man har en kvantecomputer.

Udregninger ved hjælp af kvantefysikken

Men hvordan fungerer kvantecomputere så? En almindelig computer fungerer ved at sammensætte en stribe af det man kunne kalde 'logiske skridt'. Et eksempel på et logisk skridt kunne være at sammenligne to bincære tal (dvs. tallene kun kan være '1' eller '0'). Hvis de begge er 1, så gør ikke noget, men hvis de er '00', '10' eller '01', så tænd for en lille lampe. I Figur 2 kan man se hvordan sådan et logisk skridt ser ud i praksis i et eksempel hvor det bruges i en tyverialarm. Det virker måske abstrakt og samtidigt banalt, men hvis man kan udføre dette logiske skridt (navnet på dette skridt er "negativ-og"), og man kan sætte det sammen med mange flere negativ-og'er, så har man faktisk alt hvad man skal bruge til at bygge en computer. Der findes selvfølgelig flere logiske skridt end dette ene, men det er nok i sig selv.

Men det viser sig at hvis man bruger en særlig type logiske skridt som ikke er beskrevet med klassisk logik, men med elementer fra kvantefysikken, kan man løse nogle langt mere komplekse problemer. Vi kalder den slags skridt for kvantelogiske skridt eller kvante-operationer. Hvis man bygger en computer som ikke kun har de klassiske logiske skridt, men også de skridt kvantefysikken tillader, så åbner man op for BQP-landet på problem-verdenskortet. Ligesom da Stillehavet først blev opdaget af den vestlige verden af Magellan i 1521, og man ikke vidste præcis hvor grænserne af det enorme ocean lå, ved vi endnu ikke præcist hvor grænserne for BQP-landet ligger. Vi ved, lidt ligesom med opdagelsen af sydhavsøerne, at der eksisterer nogle interessante og vigtige områder i BQP, men hele landkortet og grænserne for BQP er endnu ikke afdækket. En ting er dog helt sikkert: Der er vigtige og store problemer som ligger i NP (umulige at løse effektivt på en klassisk supercomputer) men som også ligger i BQP (nemme på en kvantecomputer), og de problemer er selvfølgelig hamrende interessante.

> ***Hvis nogen giver dig en kvantecomputer, hvordan kan du så tjekke at den gør hvad den skal?***

> ***Vi ved, lidt ligesom med opdagelse af sydhavsøerne, at der eksisterer nogle interessante og vigtige områder i BQP, men hele landkortet af grænserne for BQP er endnu ikke afdækket***

Figur 3
(a) Et billede af det eksperimentelle udstyr der er brugt i dette forskningsprojekt til at køle kvantechippen ned til lige over det absolutte nulpunkt. (b) Et elektronmikroskopi-billede af den kvantechip der (blandt mange andre) er blevet studeret i dette projekt. De fem små X'er svarer til fem små kvante-bits, som vi bruger til at implementere vores BQP-problem. (c) Et billede af Morten Kjaergaard i laboratoriet hvor han styrer de fem kvante-bit fra Figur (b).
Foto: (a & c) Nathan Fiske

Hvor langt er vi med kvantecomputeren?
Adskillige firmaer i verden har allerede bygget kvantecomputere, men de er endnu ikke store nok til at tackle de 'grand challenges' vi drømmer om. I min forskning har jeg udviklet en specialbygget kvantecomputer i vores laboratorium til at studere et helt særligt problem fra BQP. Figur 3 viser et billede af den store fryser som bruges til at køle selve kvantechippen ned til 0.01 grader over det absolutte nulpunkt (-273,15 C), og i Figur 3(b) vises et billede af en kvantechip med fem såkaldte kvantebits. Både fryseren og chippen er benyttet i dette projekt. Figur 3(c) er mig i laboratoriet med udstyr der bruges til at styre kvantecomputeren.

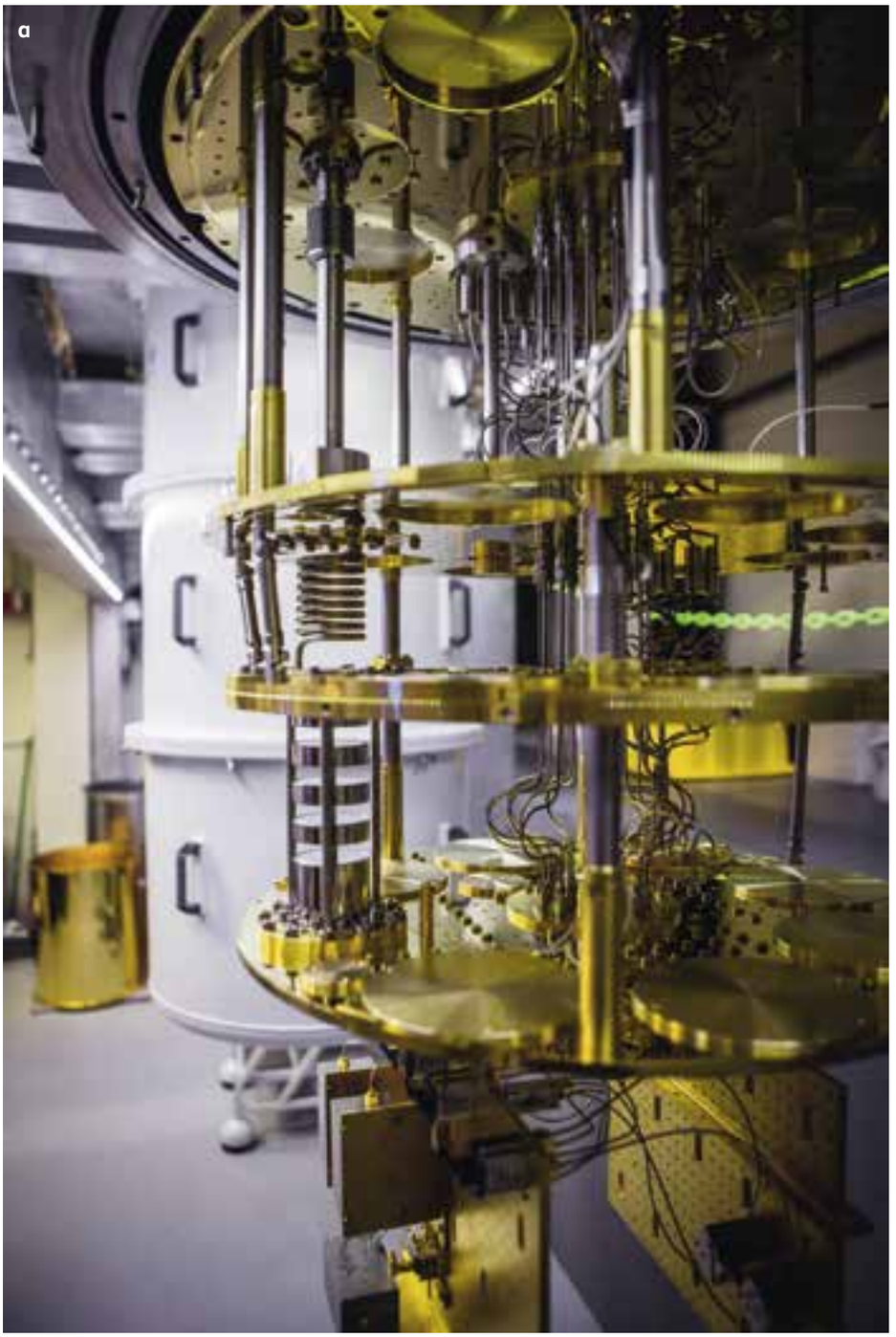
Problemet jeg har arbejdet på, er i BQP, men udenfor NP (markeret med en stjerne i Figur 1), og omhandler følgende spørgsmål: Hvis nogen giver dig en kvantecomputer, hvordan kan du så tjekke at den gør hvad den skal? Siden visse dele af BQP er uden for NP, er det altså ikke altid muligt at dobbelttjekke alle udregninger fra en kvantecomputer ved hjælp af normale computere. Det program jeg har arbejdet på, gør et særligt stort problem inden for kvantecomputervidenskab simplere. Problemet har det tekniske navn kvantetilstandstomografi, og kvanteprogrammets fulde navn er 'Density Matrix Exponentiation' og blev først foreslået i 2014 af Seth Lloyd, som er professor ved Massachusetts Institute of Technology.[3] I tæt samarbejde med professor Lloyd har jeg videreudviklet hans program i en grad som gør at vi nu prøver at køre programmet på den kvanteprocessor jeg også har arbejdet på. Vi håber at kunne fremvise programmet i løbet af 2019.

Så hvorfor findes der ikke allerede store kvantecomputere? Svaret er simpelt: Alle de kvantecomputere der findes i dag, er meget følsomme over for støj fra omgivelserne. De mindste lyspartikler eller temperaturforstyrrelser kan ødelægge en kvanteudregning. Der findes såkaldt 'kvante-fejlretningsprogrammer', som er designet til at modvirke støj og fejl under en udregning. Som et led i mit forskningsprojekt støttet af Carlsbergfondet har jeg også, i tæt samarbejde med professor Nikolaj Zinner og en studerende fra Aarhus Universitet, udviklet et kvantefejlretningsprogram som kombinerer ideer fra både teoretisk ideelle og eksperimentelt støjfyldte kvantecomputere. I fremtiden håber vi at demonstrere hvordan vores fejlretningsprogram nedsætter mængden af støj i en kvantecomputer.

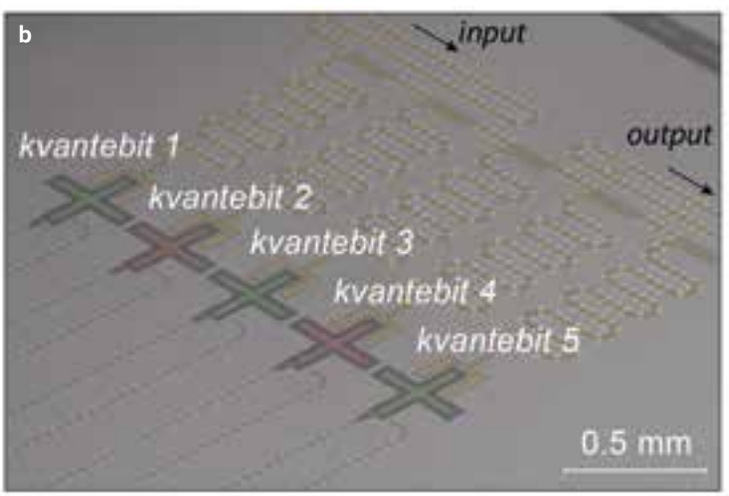
Referencer
1 A.M. Turing, "On Computable Numbers, with an Application to the Entscheidungsproblem". Proc. of the London Mathematical Society, 2, 42 (1): 230-65 (1937). **2** E. Bernstein og U. Vazirani, "Quantum Complexity Theory". SIAM Journal on Computing. 26 (5): 1411-1473 (1997). **3** S. Lloyd et al, "Quantum Principal Component Analysis". Nature Physics. 10: 631-633 (2014).

> ***Der er vigtige og store problemer som ligger i NP (umulige at løse effektivt på en klassisk supercomputer) men som også ligger i BQP (nemme på en kvantecomputer), og de problemer er selvfølgelig hamrende interessante!***

Logiske skridt
Måske virker logiske skridt meget abstrakte, men dem bruger man hele tiden i hverdagen, uden rigtigt at tænke over det. Som eksempel, kan en 'negativ-og' bruges i en tyverialarm! Forestil dig f.eks en tyverialarm sat på din dør og på vinduet i dit hus. Hvis vinduet er intakt, kan vi sige, at det svarer til '1' (og hvis det er ødelagt, kalder vi det '0'), hvis døren er intakt svarer det også til '1' (og modsat til '0', hvis den er revet op). Så når både dør og vindue er intakte, er tyverialarmen i '11' situation, og negativ-og lampen er slukket. Men hvis en tyv ødelægger vinduet bliver tyverialarm-systemet til '10' og den lille negativ-og lampe lyser præcis op! Hvis tyvene bryder ind gennem døren, bliver tyverialarmen til '01' og alarmlampen lyser også. Hvis nu tyvene både ødelægger vinduet og døren, er tyverialarmen '00' og lampen lyser også. Så et negativ-og logisk skridt kan bruges til at tænde for en tyverialarm, hvis der er indbrud.


a


c


b
input
output
kvantebit 1
kvantebit 2
kvantebit 3
kvantebit 4
kvantebit 5
0.5 mm