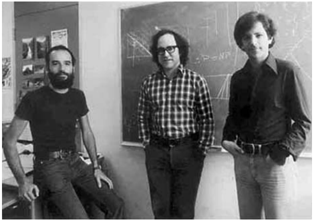
Quantum Computing: basic principles, present architectures, future possibilities

Zoltán Zimborás

GPU Days
WIGNER RCP June 21, 2018.

Finding the prime factors of integers is hard

- The most popular public-key cryptosystem, the RSA (Rivest-Shamir-Adleman) encryption, which was developed already in 1978, uses the observation that multiplying integers is easy, factoring integers into prime factors is hard.



- For example, let us have a look at the factors of the following 232 decimal digits (768 bits) number

```
RSA-768 = 12301866845301177551304949583849627207728535695953347921973224521517264005
          07263657518745202199786469389956474942774063845925192557326303453731548268
          50791702612214291346167042921431160222124047927473779408066535141959745985
          6902143413
```

```
RSA-768 = 33478071698956898786044169848212690817704794983713768568912431388982883793
          878002287614711652531743087737814467999489
        × 36746043666799590428244633799627952632279158164343087642676032283815739666
          511279233373417143396810270092798736308917
```

The RSA Factoring Challenge

- What about the following 230 decimal digits (762 bits) number?

```
RSA-232 = 1009881397871923546909564894309468582818233821955573955141120516205831021338
          5285453743661097571543636649133800849170651699217015247332943892702802343809
          6090980497644054071120196541074755382494867277137407501157718230539834060616
          2079
```

RSA number	Decimal digits	Binary digits	Cash prize offered	Factored on	Factored by
RSA-100	100	330	US$1,000[4]	April 1, 1991[5]	Arjen K. Lenstra
RSA-110	110	364	US$4,429[4]	April 14, 1992[5]	Arjen K. Lenstra and M.S. Manasse
RSA-120	120	397	$5,898[4]	July 9, 1993[6]	T. Denny *et al.*
RSA-129 [**]	129	426	$100 USD	April 26, 1994[5]	Arjen K. Lenstra *et al.*
RSA-130	130	430	US$14,527[4]	April 10, 1996	Arjen K. Lenstra *et al.*
RSA-140	140	463	US$17,226	February 2, 1999	Herman te Riele *et al.*
RSA-150	150	496		April 16, 2004	Kazumaro Aoki *et al.*
RSA-155	155	512	$9,383[4]	August 22, 1999	Herman te Riele *et al.*
RSA-160	160	530		April 1, 2003	Jens Franke *et al.*, University of Bonn
RSA-170 [*]	170	563		December 29, 2009	D. Bonenberger and M. Krone [***]
RSA-576	174	576	$10,000 USD	December 3, 2003	Jens Franke *et al.*, University of Bonn
RSA-180 [*]	180	596		May 8, 2010	S. A. Danilov and I. A. Popovyan, Moscow State University[7]
RSA-190 [*]	190	629		November 8, 2010	A. Timofeev and I. A. Popovyan
RSA-640	193	640	$20,000 USD	November 2, 2005	Jens Franke *et al.*, University of Bonn
RSA-200 [*] ?	200	663		May 9, 2005	Jens Franke *et al.*, University of Bonn
RSA-210 [*]	210	696		September 26, 2013[8]	Ryan Propper
RSA-704 [*]	212	704	$30,000 USD	July 2, 2012	Shi Bai, Emmanuel Thomé and Paul Zimmermann
RSA-220 [*]	220	729		May 13, 2016	S. Bai, P. Gaudry, A. Kruppa, E. Thomé and P. Zimmermann
RSA-230	230	762			
RSA-232	232	768			
RSA-768 [*]	232	768	$50,000 USD	December 12, 2009	Thorsten Kleinjung *et al.*
RSA-240	240	795			
RSA-250	250	829			
RSA-260	260	862			
RSA-270	270	895			
RSA-896	270	896	$75,000 USD		
RSA-280	280	928			
RSA-290	290	962			
RSA-300	300	995			
RSA-309	309	1024			
RSA-1024	309	1024	$100,000 USD		
RSA-310	310	1028			
RSA-320	320	1061			
RSA-330	330	1094			
RSA-340	340	1128			
RSA-350	350	1161			
RSA-360	360	1194			
RSA-370	370	1227			
RSA-380	380	1261			
RSA-390	390	1294			
RSA-400	400	1327			
RSA-410	410	1360			
RSA-420	420	1393			
RSA-430	430	1427			
RSA-440	440	1460			
RSA-450	450	1493			
RSA-460	460	1526			
RSA-1536	463	1536	$150,000 USD		
RSA-470	470	1559			
RSA-480	480	1593			
RSA-490	490	1626			
RSA-500	500	1659			
RSA-617	617	2048			
RSA-2048	617	2048	$200,000 USD		

How hard is it to break RSA?

How much computing resource is required to brute-force RSA?

17

11

It's been over 30 years since Rivest, Shamir and Adleman first publicly described their algorithm for public-key cryptography; and the intelligence community is thought to have known about it for around 40 years—possibly longer.

It's fair to assume that, during those 40 years, certain three-letter organisations have employed their vast resources toward "breaking" RSA. One brute-force approach may have been to enumerate every possible key-pair such that, upon encountering a message known to be encrypted with a particular public-key, they need merely lookup the associated private-key in order to decrypt that message. Signatures could be forged similarly.

How reasonable is this hypothesis? How much computing resource would have been required over those 40 years to enumerate every possible {1024,2048,4096}-bit key-pair? I think it best to avoid discussion and leave the question of whether the spooks could have harnessed such resource as an exercise to the reader.

cryptanalysis public-key rsa brute-force-attack

share improve this question

asked Jun 25 '12 at 6:14
eggyal
232 1 2 10

asked 5 years, 11 months ago

viewed 27,215 times

active 2 years, 11 months ago

INTERNATIONAL ASSOCIATION FOR CRYPTOLOGIC RESEARCH
i a C r

How hard is it to break RSA?

20

It's not possible.

The number of primes smaller than x is approximately $\frac{x}{\ln x}$. Therefore the number of 512 bit primes (approximately the length you need for 1024 bit modulus) is approximately:

$$\frac{2^{513}}{\ln 2^{513}} - \frac{2^{512}}{\ln 2^{512}} \approx 2.76 \times 10^{151}$$

The number of RSA moduli (i.e. pair of two distinct primes) is therefore:

$$\frac{(2.76 \times 10^{151})^2}{2} - 2.76 \times 10^{151} = 1.88 \times 10^{302}$$

Now consider that the observable universe contains about 10^{80} atoms. Assume that you could use each of those atoms as a CPU, and each of those CPUs could enumerate one modulus per millisecond. To enumerate all 1024 bit RSA moduli you would need:

$$\begin{aligned} 1.88 \times 10^{302} ms/10^{80} &= 1.88 \times 10^{222} ms \\ &= 1.88 \times 10^{219} s \\ &= 5.22 \times 10^{215} h \\ &= 5.95 \times 10^{211} \text{years} \end{aligned}$$

Just as a comparison: the universe is about 13.75×10^9 years old.

It's not a question of resources, it's simply not possible.

Also, it would not make any sense to do that. There are much faster ways to find out a secret key. In fact there are algorithms with sub-exponential running time for factoring integers.

International Journal of Theoretical Physics, Vol. 21, Nos. 6/7, 1982

Simulating Physics with Computers

Richard P. Feynman

Department of Physics, California Institute of Technology, Pasadena, California 91107

Received May 7, 1981

1. INTRODUCTION

On the program it says this is a keynote speech—and I don't know what a keynote speech is. I do not intend in any way to suggest what should be in this meeting as a keynote of the subjects or anything like that. I have my own things to say and to talk about and there's no implication that anybody needs to talk about the same thing or anything like it. So what I want to talk about is what Mike Dertouzos suggested that nobody would talk about. I want to talk about the problem of simulating physics with computers and I mean that in a specific way which I am going to explain. The reason for doing this is something that I learned about from Ed Fredkin, and my entire interest in the subject has been inspired by him. It has to do with learning something about the possibilities of computers, and also something about possibilities in physics. If we suppose that we know all the physical laws perfectly, of course we don't have to pay any attention to computers. It's interesting anyway to entertain oneself with the idea that we've got something to learn about physical laws; and if I take a relaxed view here (after all I'm here and not at home) I'll admit that we don't understand everything.

The first question is, What kind of computer are we going to use to simulate physics? Computer theory has been developed to a point where it realizes that it doesn't make any difference; when you get to a *universal computer*, it doesn't matter how it's manufactured, how it's actually made. Therefore my question is, Can physics be simulated by a universal computer? I would like to have the elements of this computer *locally interconnected*, and therefore sort of think about cellular automata as an example (but I don't want to force it). But I do want something involved with the

locality of interaction. I would not like to think of a very enormous computer with arbitrary interconnections throughout the entire thing.

Now, what kind of physics are we going to imitate? First, I am going to describe the possibility of simulating physics in the classical approximation, a thing which is usually described by local differential equations. But the physical world is quantum mechanical, and therefore the proper problem is the simulation of quantum physics—which is what I really want to talk about, but I'll come to that later. So what kind of simulation do I mean? There is, of course, a kind of approximate simulation in which you design numerical algorithms for differential equations, and then use the computer to compute these algorithms and get an approximate view of what physics ought to do. That's an interesting subject, but is not what I want to talk about. I want to talk about the possibility that there is to be an *exact* simulation, that the computer will do *exactly* the same as nature. If this is to be proved and the type of computer is as I've already explained, then it's going to be necessary that *everything* that happens in a finite volume of space and time would have to be exactly analyzable with a finite number of logical operations. The present theory of physics is not that way, apparently. It allows space to go down into infinitesimal distances, wavelengths to get infinitely great, terms to be summed in infinite order, and so forth; and therefore, if this proposition is right, physical law is wrong.

So good, we already have a suggestion of how we might modify physical law, and that is the kind of reason why I like to study this sort of problem. To take an example, we might change the idea that space is continuous to the idea that space perhaps is a simple lattice and everything is discrete (so that we can put it into a finite number of digits) and that time jumps discontinuously. Now let's see what kind of a physical world it would be or what kind of problem of computation we would have. For example, the first difficulty that would come out is that the speed of light would depend slightly on the direction, and there might be other anisotropies in the physics that we could detect experimentally. They might be very small anisotropies. Physical knowledge is of course always incomplete, and you can always say we'll try to design something which beats experiment at the present time, but which predicts anistropies on some scale to be found later. That's fine. That would be good physics if you could predict something consistent with all the known facts and suggest some new fact that we didn't explain, but I have no specific examples. So I'm not objecting to the fact that it's anistropic in principle, it's a question of how anistropic. If you tell me it's so-and-so anistropic, I'll tell you about the experiment with the lithium atom which shows that the anistropy is less than that much, and that this here theory of yours is impossible.

You cannot even describe the state of 100 quantum dipole moments (spins) with any future classical computer. What should we do?

Feynman's proposal

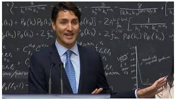
Richard Feynman (1981):

"...trying to find a computer simulation of physics, seems to me to be an excellent program to follow out...and I'm not happy with all the analyses that go with just the classical theory, because *nature isn't classical*, dammit, and if you want to make a simulation of nature, you'd better *make it quantum mechanical*, and by golly it's a wonderful problem because it doesn't look so easy."

This opened the way for the idea of quantum algorithms (Deutsch '85, Shor '94, Grover '96)

Recent buzz around quantum computing

- Quantum Computing is very popular nowadays:
  - Everybody talks about this
    from the Canadian Prime Minister to EU officials.

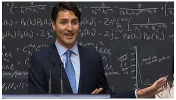


  - Recent Nobel prize given to related research (Haroche, Wineland).

  - Many physicists specializing in this field get jobs in Multinational Companies .
  - EU Quantum Technology Flagship, US Quantum Technology Strategy
  - Invited talks in GPU Days

Recent buzz around quantum computing

- Quantum Computing is very popular nowadays:
    - Everybody talks about this
      from the Canadian Prime Minister to EU officials.

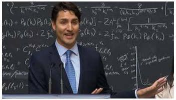


- Recent Nobel prize given to related research (Haroche, Wineland).

- Many physicists specializing in this field get jobs in Multinational Companies .
- EU Quantum Technology Flagship, US Quantum Technology Strategy
- Invited talks in GPU Days

Recent buzz around quantum computing

- Quantum Computing is very popular nowadays:
  - Everybody talks about this
    from the Canadian Prime Minister to EU officials.

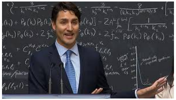


  - Recent Nobel prize given to related research (Haroche, Wineland).

  - Many physicists specializing in this field get jobs in Multinational Companies .
  - EU Quantum Technology Flagship, US Quantum Technology Strategy
  - Invited talks in GPU Days

Recent buzz around quantum computing

- Quantum Computing is very popular nowadays:
  - Everybody talks about this
    from the Canadian Prime Minister to EU officials.

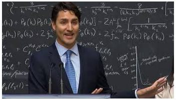


  - Recent Nobel prize given to related research (Haroche, Wineland).

  - Many physicists specializing in this field get jobs in Multinational Companies .
  - EU Quantum Technology Flagship, US Quantum Technology Strategy
  - Invited talks in GPU Days

Recent buzz around quantum computing

- Quantum Computing is very popular nowadays:
  - Everybody talks about this
    from the Canadian Prime Minister to EU officials.



  - Recent Nobel prize given to related research (Haroche, Wineland).

  - Many physicists specializing in this field get jobs in Multinational Companies .
  - EU Quantum Technology Flagship, US Quantum Technology Strategy
  - Invited talks in GPU Days

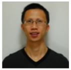
Google created already two types of Quantum Engineer positions

PI

John M. Martinis
Professor at UC Santa Barbara since 2004
Research Scientist at Google since 2014
martinis (at) physics (dot) ucsb (dot) edu
jmartinis (at) google (dot) com

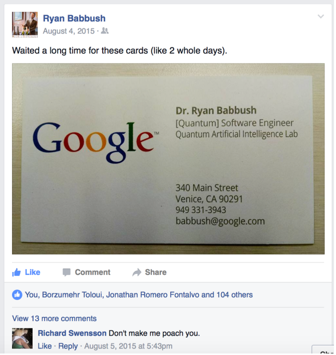
Quantum Electronics Engineers, Google

Rami Barends
Post Doctoral Fellow, 2010-2014
Quantum Electronics Engineer at Google since 2014
rbarends (at) physics (dot) ucsb (dot) edu
barends (at) google (dot) com



Yu Chen
Post Doctoral Fellow, 2010-2014
Quantum Electronics Engineer at Google since 2014
ychen (at) physics (dot) ucsb (dot) edu
bryanchen (at) google (dot) com

Austin Fowler
Staff Scientist, 2013-2014
Quantum Electronics Engineer at Google since 2014
agfowler (at) google (dot) com



Lots of quantum start-ups

Company	Date initiated	Area	Affiliate University or Research Institute	Headquarters
1QBit	1 December 2012	Computing		Vancouver, Canada
Accenture[1]	14 June 2017	Computing		
imec[2]		Silicon Quantum Computing		Belgium
Airbus[3]	2015	Computing		Blagnac, France
Aliyun (Alibaba Cloud)[4]	30 July 2015	Computing/Communication[4][5]	Chinese Academy of Sciences [6][5][7]	Hangzhou, China
AT&T[8]	2011	Communication		Dallas, TX, USA
Atos[9]		Communication		Bezons, France
Booz Allen Hamilton[10]		Computing		Tysons Corner, VA, USA
BT[11]		Communication		London, UK
Carl Zeiss AG[12]			University College London	Oberkochen, Germany
Cambridge Quantum Computing Limited[13]		Communication		Cambridge, UK
D-Wave	1 January 1999	Computing		Burnaby, Canada
Fujitsu[14]	28 September 2015	Communication	University of Tokyo	Tokyo, Japan
Google QuAIL[15]	16 May 2013	Computing	UCSB	Mountain View, CA, USA
HP[16][17]		Computing[16]/Communication[17]		Palo Alto, CA, USA
Hitachi		Computing	University of Cambridge, University College London	Tokyo, Japan
Honeywell[18][19]		Computing	Georgia Tech,[18] University of Maryland[19]	Morris Plains, NJ, USA
HRL Laboratories		Computing		Malibu, CA, USA
Huawei Noah's Ark Lab[20]		Communication	Nanjing University	Shenzhen, China
IBM[21]	10 September 1990[22]	Computing	MIT[23]	Armonk, NY, USA
ID Quantique	1 July 2001	Communication		Geneva, Switzerland
ionQ[24][25]		Computing	University of Maryland, Duke University	College Park, MD, USA
Intel[26]	3 September 2015	Computing	TU Delft	Santa Clara, CA, USA
KPN[27]		Communication		The Hague, Netherlands
Lockheed Martin		Computing	University of Southern California, University College London	Bethesda, MD, USA
MagiQ		Communication		Somerville, MA, USA
Microsoft Research QuArC	19 December 2011	Computing	TU Delft, Niels Bohr Institute, University of Sydney, Purdue University, University of Maryland, ETH Zurich, UCSB	Redmond, WA, USA
Microsoft Research Station Q	22 April 2005	Computing	UCSB	Santa Barbara, CA, USA
Mitsubishi[28]		Communication		Tokyo, Japan
NEC Corporation[29]	29 April 1999[30]	Communication	University of Tokyo	Tokyo, Japan
Nokia Bell Labs[31][32]		Computing	University of Oxford	Murray Hill, NJ, USA
Northrop Grumman		Computing		West Falls Church, VA, USA
NTT Laboratories[33]		Computing	Bristol University	Tokyo, Japan
Q-Ctrl[34][35][36]	2017	Computing[note 1]		Sydney, Australia

Early opinions

QUANTUM COMPUTING: DREAM OR NIGHTMARE?

Recent experiments have deepened our insight into the wonderfully counterintuitive quantum theory. But are they really harbingers of quantum computing? We doubt it.

Serge Haroche and Jean-Michel Raimond

The principles of quantum computing were laid out about 15 years ago by computer scientists applying the superposition principle of quantum mechanics to computer operation. Quantum computing has recently become a hot topic in physics, with the recognition that a two-level system can be presented as a quantum bit, or "qubit," and that an interaction between such systems could lead to the building of quantum gates obeying nonclassical logic. (See PHYSICS TODAY, October 1995, page 24 and March 1996, page 21.)

two interacting qubits: a "control" bit and a "target" bit. The control remains unchanged, but its state determines the evolution of the target: If the control is 0, nothing happens to the target; if it is 1, the target undergoes a well-defined transformation.

Quantum mechanics admits additional options. If the control is in some coherent superposition of 0 and 1, the output of the gate is entangled. That is to say, the two qubits are strongly correlated in a nonseparable state, analogous to the particle pairs of the Einstein–Podolsky–Rosen paradox. The

Early opinions

brothers. How can we get kids excited about becoming scientists, engineers, or technological entrepreneurs if they are taught a form of history in which role models are removed?

Under the Dole administration, I look forward to working with you in an era where good science will be consistently supported.

ROBERT J. DOLE
Washington, DC

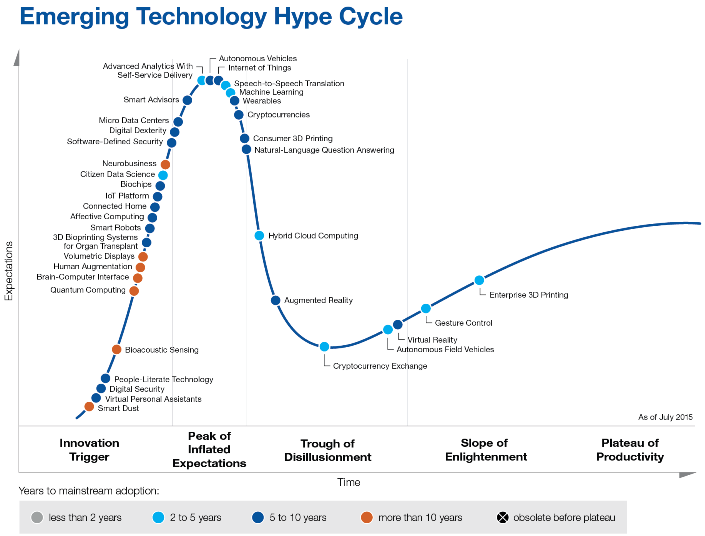
Future of Quantum Computing Proves to Be Debatable

In presenting their opinions in the article "Quantum Computing: Dream or Nightmare?" (August, page 51), Serge Haroche and Jean-Michel Raimond conclude that large-scale quantum computation will remain merely a dream of computer theorists. Their principal argument is that, for a quantum computer to be would be useful only if R is of order 10^{11}, or that any application requiring more than 3×10^6 optical operations would be fundamentally disallowed.

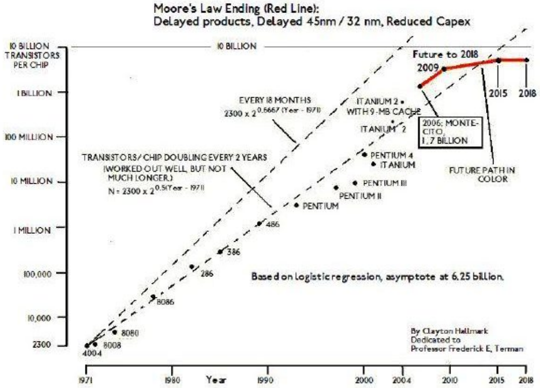
Experimentally, our laboratory has demonstrated a "controlled-NOT" quantum logic gate with a single trapped ion,[4] following the ideas of Ignacio Cirac and Peter Zoller.[5] (See PHYSICS TODAY, March, page 21.) In the experiment, R was about 10^1 and the gate time was about 50 s. However, as is often the case in experimental physics, this apparatus was assembled with the least effort necessary to exhibit the desired behavior and should not be taken to represent the technological limit. Although the task of scaling this system to large numbers of ions and gates involving massively entangled quantum states is daunting, the pitfalls are technical, not fundamental.

It is too early to make absolute assertions regarding the viability of quantum computation when such a large degree of uncertainty in both

The (trivial) emerging technology hype cycle



Moore's Law


Moore's Law Ending (Red Line):
Delayed products, Delayed 45nm / 32 nm, Reduced Capex
10 BILLION TRANSISTORS PER CHIP
10 BILLION
1 BILLION
100 MILLION
10 MILLION
1 MILLION
100,000
10,000
2300
Future to 2018
2009
2015
2018
EVERY 18 MONTHS
2300 x 2^0.6667 (Year - 1971)
ITANIUM 2 WITH 9-MB CACHE
ITANIUM 2
2006: MONTE-CITO, 1.7 BILLION
TRANSISTORS/CHIP DOUBLING EVERY 2 YEARS (WORKED OUT WELL, BUT NOT MUCH LONGER.)
N = 2300 x 2^0.5(Year - 1971)
PENTIUM 4
ITANIUM
FUTURE PATH IN COLOR
PENTIUM III
PENTIUM II
PENTIUM
486
386
286
8086
8080
8008
4004
Based on logistic regression, asymptote at 6.25 billion.
By Clayton Hallmark
Dedicated to
Professor Frederick E. Terman
1971
1980
Year
1990
2000
2004
2010
2015
2018

Outline of the rest of talk

- Principles of Quantum Computing (Quantum Parallelism and the Gate Model)
- Two architecture types: the Gate Model and Adiabatic Quantum Computing
- What can a Quantum Computer do that a Classical Computer cannot? Classical and Quantum Complexity Theory.
- What are the future perspectives?

Outline of the rest of talk

- Principles of Quantum Computing (Quantum Parallelism and the Gate Model)
- Two architecture types: the Gate Model and Adiabatic Quantum Computing
- What can a Quantum Computer do that a Classical Computer cannot? Classical and Quantum Complexity Theory.
- What are the future perspectives?

Outline of the rest of talk

- Principles of Quantum Computing (Quantum Parallelism and the Gate Model)
- Two architecture types: the Gate Model and Adiabatic Quantum Computing
- What can a Quantum Computer do that a Classical Computer cannot? Classical and Quantum Complexity Theory.
- What are the future perspectives?

Outline of the rest of talk

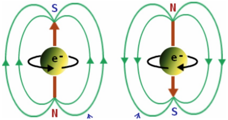
Zoltán Zimborás

- Principles of Quantum Computing (Quantum Parallelism and the Gate Model)
- Two architecture types: the Gate Model and Adiabatic Quantum Computing
- What can a Quantum Computer do that a Classical Computer cannot? Classical and Quantum Complexity Theory.
- What are the future perspectives?

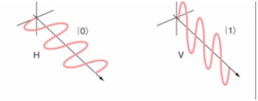
Outline of the rest of talk

- Principles of Quantum Computing (Quantum Parallelism and the Gate Model)
- Two architecture types: the Gate Model and Adiabatic Quantum Computing
- What can a Quantum Computer do that a Classical Computer cannot? Classical and Quantum Complexity Theory.
- What are the future perspectives?

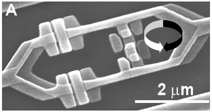
The magnetic dipole moment of an electron (or a nucleus):



The polarization of light A fény polarizációja:



The flux or the direction of current in superconducting rings:

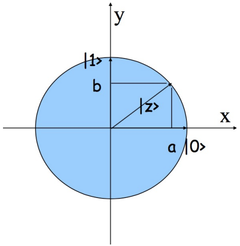


The principle of superposition

According to the principle of superposition, the general state of a qubit is

$$|z\rangle = a|0\rangle + b|1\rangle.$$

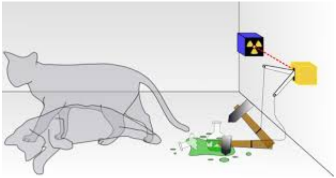
Here $|a|^2$ provides the probability that we find the system to be in state $|0\rangle$ when measured, and $|b|^2$ provides the probability that we find it to be in state $|1\rangle$. We have to assume that $|a|^2 + |b|^2 = 1$

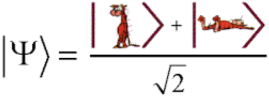


The most famous qubit

The most famous qubit

Schrödinger's cat

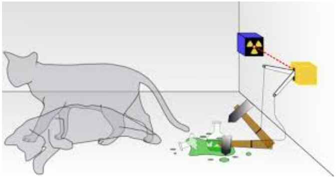


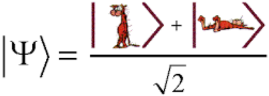


But what is the difference between $a|0\rangle + b|1\rangle$ and $a|0\rangle - b|1\rangle$?

The most famous qubit

Schrödinger's cat





But what is the difference between $a|0\rangle + b|1\rangle$ and $a|0\rangle - b|1\rangle$?

$$V|0\rangle = a|0\rangle + b|1\rangle,$$
$$V|1\rangle = c|0\rangle + d|1\rangle,$$
$$V|z\rangle = V(e|0\rangle + f|1\rangle) = eV|0\rangle + fV|1\rangle \quad = (ea + fc)|0\rangle + (eb + fd)|1\rangle.$$

We can gather the above numbers in matrix:

$$V = \begin{pmatrix} a & c \\ b & d \end{pmatrix}.$$

Similarly, we could introduce n qubit states (and the respective operations):

$$q|0\rangle|0\rangle + r|0\rangle|1\rangle + s|1\rangle|0\rangle + t|1\rangle|1\rangle$$

Allowed operations on qubits

$$V|0\rangle = a|0\rangle + b|1\rangle,$$
$$V|1\rangle = c|0\rangle + d|1\rangle,$$
$$V|z\rangle = V(e|0\rangle + f|1\rangle) = eV|0\rangle + fV|1\rangle \quad = (ea + fc)|0\rangle + (eb + fd)|1\rangle.$$

We can gather the above numbers in matrix:

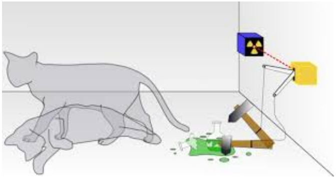
$$V = \begin{pmatrix} a & c \\ b & d \end{pmatrix}.$$

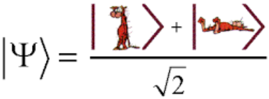
Similarly, we could introduce n qubit states (and the respective operations):

$$q|0\rangle|0\rangle + r|0\rangle|1\rangle + s|1\rangle|0\rangle + t|1\rangle|1\rangle$$

The most famous qubit

Schrödinger's cat

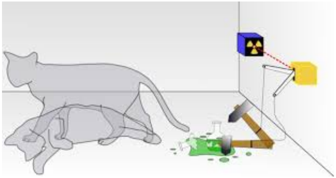


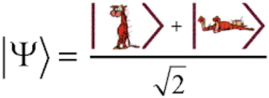


The surprise in Schrödinger's thought experiment is not that with 50% probability the cat is alive and with 50% it is dead, rather the fact that there exists a resurrection operator. (Reinhard Werner)

The most famous qubit

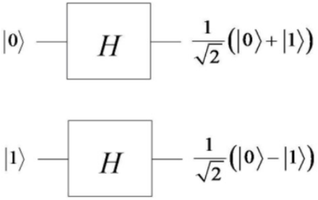
Schrödinger's cat





The surprise in Schrödinger's thought experiment is not that with 50% probability the cat is alive and with 50% it is dead, rather the fact that there exists a resurrection operator. (Reinhard Werner)

The Hadamard gate



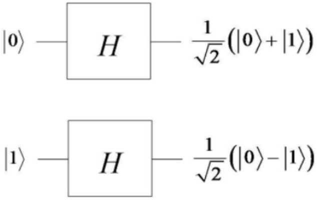
How does such a gate act on a Schrödinger cat state?

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

How does such a gate act on the alternative Schrödinger cat state?

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

The Hadamard gate



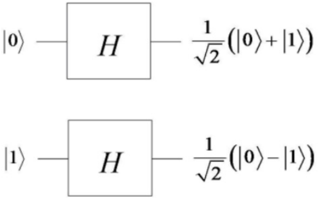
How does such a gate act on a Schrödinger cat state?

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

How does such a gate act on the alternative Schrödinger cat state?

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

The Hadamard gate



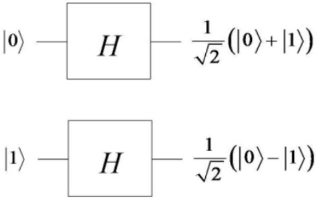
How does such a gate act on a Schrödinger cat state?

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

How does such a gate act on the alternative Schrödinger cat state?

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

The Hadamard gate



How does such a gate act on a Schrödinger cat state?

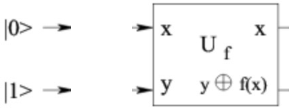
$$H\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) = \frac{1}{2}|0\rangle + \frac{1}{2}|1\rangle + \frac{1}{2}|0\rangle - \frac{1}{2}|1\rangle = |0\rangle.$$

How does such a gate act on the alternative Schrödinger cat state?

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

Deutsch's problem

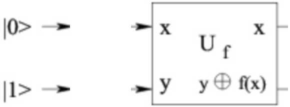
Let f be a Boole functions that maps a single bit into a single bit. With how many trials (or queries of f) can we decide whether it is a constant function or not?



Obviously with two.

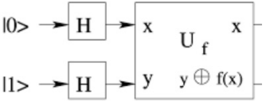
Deutsch's problem

Let f be a Boole functions that maps a single bit into a single bit. With how many trials (or queries of f) can we decide whether it is a constant function or not?



Obviously with two.

We can also insert a superposition

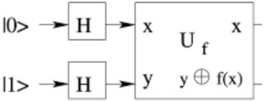


The answer is somehow included in resulting state

$$\frac{1}{4}|0\rangle|1{+}f(0)\rangle + \frac{1}{4}|1\rangle|1{+}f(1)\rangle + \frac{1}{4}|0\rangle|f(0)\rangle - \frac{1}{4}|1\rangle|f(1)\rangle.$$

But how can we obtain the answer from the state?

We can also insert a superposition

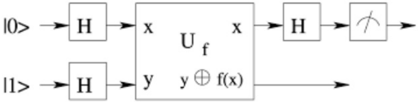


The answer is somehow included in resulting state

$$\frac{1}{4}|0\rangle|1+f(0)\rangle + \frac{1}{4}|1\rangle|1+f(1)\rangle + \frac{1}{4}|0\rangle|f(0)\rangle - \frac{1}{4}|1\rangle|f(1)\rangle.$$

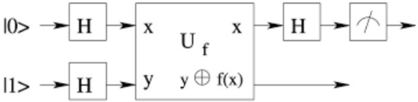
But how can we obtain the answer from the state?

Let us act with another Hadamard gate



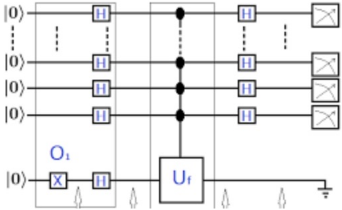
The first qubit of the resulting state is with 100% probability in state $|0\rangle$ if f is constant, while it is in state $|1\rangle$ if f is not constant. One query/trial is enough!

Let us act with another Hadamard gate

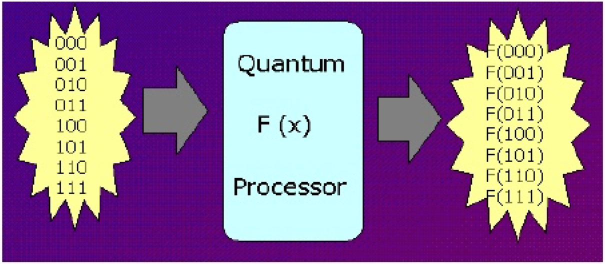


The first qubit of the resulting state is with 100% probability in state $|0\rangle$ if f is constant, while it is in state $|1\rangle$ if f is not constant. One query/trial is enough!

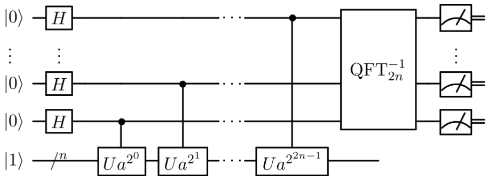
Generalizing the problem to Boole functions with many variables



Naive quantum parallelism



The Shor algorithm



$$15 = 3 \cdot 5 \quad (2001)$$

$$143 = 11 \cdot 13 \quad (2012)$$

$$56153 = 241 \cdot 233 \quad (2014)$$

RSA-640 [edit]

RSA-640 has 640 bits (193 decimal digits). A cash prize of US$20,000 was offered by RSA Security for a successful factorization. On November 2, 2005, F. Bahr, M. Boehm, J. Franke and T. Kleinjung of the German Federal Office for Information Security announced that they had factorized the number using GNFS as follows:[25][26][27]

```
RSA-640 = 3107418240490043721350750035888567930037346022842727545 7
```

```
RSA-640 = 31074182404900437213507500358885679300373460228427275457
          20161948823206440518081504556346829671723286782437916272
          83803341547107310850191954852900733772482278352574238645
          4014691736602477652346609
```

```
RSA-640 = 16347336458092538484431338838650908598417836700330923121
          81110852389333100104508151212118167511579
        × 19008712816648221131268515739354139754718967899685154936
          66638539088027103802104498957191261465571
```

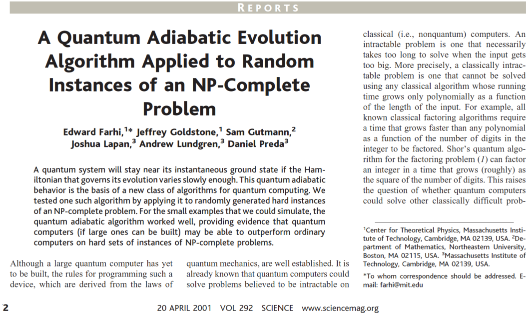
The computation took 5 months on 80 2.2 GHz AMD Opteron CPUs.

- Adiabatic theorem [M. Born, V. Fock, 1928]:
  A physical system remains in its instantaneous eigenstate if a given perturbation is acting on it slowly enough and if there is a gap between the eigenvalue and the rest of the Hamiltonian's spectrum.

- Adiabatic Quantum Computing:

$$H(t) = (1 - t/T)H_B + t/T H_P$$

$$H_B = \sum_i X_i \,, \quad H_P = \sum_i h_i Z_i + \sum_{ij} J_{ij} Z_i Z_j$$

- 

REPORTS

A Quantum Adiabatic Evolution Algorithm Applied to Random Instances of an NP-Complete Problem

Edward Farhi,[1]* Jeffrey Goldstone,[1] Sam Gutmann,[2] Joshua Lapan,[3] Andrew Lundgren,[3] Daniel Preda[3]

A quantum system will stay near its instantaneous ground state if the Hamiltonian that governs its evolution varies slowly enough. This quantum adiabatic behavior is the basis of a new class of algorithms for quantum computing. We tested one such algorithm by applying it to randomly generated hard instances of an NP-complete problem. For the small examples that we could simulate, the quantum adiabatic algorithm worked well, providing evidence that quantum computers (if large ones can be built) may be able to outperform ordinary computers on hard sets of instances of NP-complete problems.

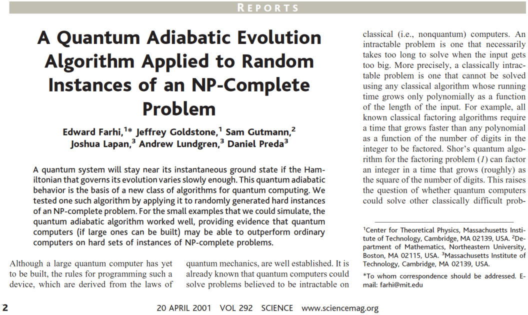
Although a large quantum computer has yet to be built, the rules for programming such a device, which are derived from the laws of quantum mechanics, are well established. It is already known that quantum computers could solve problems believed to be intractable on classical (i.e., nonquantum) computers. An intractable problem is one that necessarily takes too long to solve when the input gets too big. More precisely, a classically intractable problem is one that cannot be solved using any classical algorithm whose running time grows only polynomially as a function of the length of the input. For example, all known classical factoring algorithms require a time that grows faster than any polynomial as a function of the number of digits in the integer to be factored. Shor's quantum algorithm for the factoring problem (*1*) can factor an integer in a time that grows (roughly) as the square of the number of digits. This raises the question of whether quantum computers could solve other classically difficult prob-

[1]Center for Theoretical Physics, Massachusetts Institute of Technology, Cambridge, MA 02139, USA. [2]Department of Mathematics, Northeastern University, Boston, MA 02115, USA. [3]Massachusetts Institute of Technology, Cambridge, MA 02139, USA.

*To whom correspondence should be addressed. E-mail: farhi@mit.edu

2 20 APRIL 2001 VOL 292 SCIENCE www.sciencemag.org

- Kitaev: the adiabatic and gate models are computationally equivalent.
- Error correction seems possible for the gate model, but seems hopeless for AQC.

- 

REPORTS

A Quantum Adiabatic Evolution Algorithm Applied to Random Instances of an NP-Complete Problem

Edward Farhi,[1*] Jeffrey Goldstone,[1] Sam Gutmann,[2] Joshua Lapan,[3] Andrew Lundgren,[3] Daniel Preda[3]

A quantum system will stay near its instantaneous ground state if the Hamiltonian that governs its evolution varies slowly enough. This quantum adiabatic behavior is the basis of a new class of algorithms for quantum computing. We tested one such algorithm by applying it to randomly generated hard instances of an NP-complete problem. For the small examples that we could simulate, the quantum adiabatic algorithm worked well, providing evidence that quantum computers (if large ones can be built) may be able to outperform ordinary computers on hard sets of instances of NP-complete problems.

Although a large quantum computer has yet to be built, the rules for programming such a device, which are derived from the laws of quantum mechanics, are well established. It is already known that quantum computers could solve problems believed to be intractable on classical (i.e., nonquantum) computers. An intractable problem is one that necessarily takes too long to solve when the input gets too big. More precisely, a classically intractable problem is one that cannot be solved using any classical algorithm whose running time grows only polynomially as a function of the length of the input. For example, all known classical factoring algorithms require a time that grows faster than any polynomial as a function of the number of digits in the integer to be factored. Shor's quantum algorithm for the factoring problem (*1*) can factor an integer in a time that grows (roughly) as the square of the number of digits. This raises the question of whether quantum computers could solve other classically difficult prob-

[1]Center for Theoretical Physics, Massachusetts Institute of Technology, Cambridge, MA 02139, USA. [2]Department of Mathematics, Northeastern University, Boston, MA 02115, USA. [3]Massachusetts Institute of Technology, Cambridge, MA 02139, USA.

*To whom correspondence should be addressed. E-mail: farhi@mit.edu

2 20 APRIL 2001 VOL 292 SCIENCE www.sciencemag.org

- Kitaev: the adiabatic and gate models are computationally equivalent.
- Error correction seems possible for the gate model, but seems hopeless for AQC.

Adiabatic Quantum Computing

-

REPORTS

A Quantum Adiabatic Evolution Algorithm Applied to Random Instances of an NP-Complete Problem

Edward Farhi,[1*] Jeffrey Goldstone,[1] Sam Gutmann,[2] Joshua Lapan,[3] Andrew Lundgren,[3] Daniel Preda[3]

A quantum system will stay near its instantaneous ground state if the Hamiltonian that governs its evolution varies slowly enough. This quantum adiabatic behavior is the basis of a new class of algorithms for quantum computing. We tested one such algorithm by applying it to randomly generated hard instances of an NP-complete problem. For the small examples that we could simulate, the quantum adiabatic algorithm worked well, providing evidence that quantum computers (if large ones can be built) may be able to outperform ordinary computers on hard sets of instances of NP-complete problems.

Although a large quantum computer has yet to be built, the rules for programming such a device, which are derived from the laws of quantum mechanics, are well established. It is already known that quantum computers could solve problems believed to be intractable on classical (i.e., nonquantum) computers. An intractable problem is one that necessarily takes too long to solve when the input gets too big. More precisely, a classically intractable problem is one that cannot be solved using any classical algorithm whose running time grows only polynomially as a function of the length of the input. For example, all known classical factoring algorithms require a time that grows faster than any polynomial as a function of the number of digits in the integer to be factored. Shor's quantum algorithm for the factoring problem (*1*) can factor an integer in a time that grows (roughly) as the square of the number of digits. This raises the question of whether quantum computers could solve other classically difficult prob-

[1]Center for Theoretical Physics, Massachusetts Institute of Technology, Cambridge, MA 02139, USA. [2]Department of Mathematics, Northeastern University, Boston, MA 02115, USA. [3]Massachusetts Institute of Technology, Cambridge, MA 02139, USA.

*To whom correspondence should be addressed. E-mail: farhi@mit.edu

2 20 APRIL 2001 VOL 292 SCIENCE www.sciencemag.org

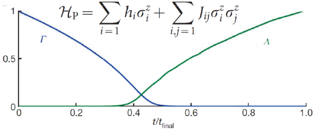
- Kitaev: the adiabatic and gate models are computationally equivalent.
- Error correction seems possible for the gate model, but seems hopeless for AQC.

DW-1: Overview

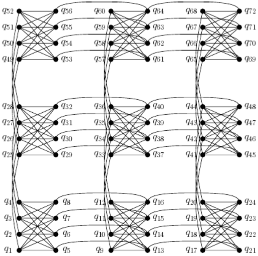
Adiabatic evolution:

$$\mathcal{H}(t) = \Gamma(t) \sum_{i=1}^{N} \Delta_i \sigma_i^x + \Lambda(t) \mathcal{H}_P$$

$$\mathcal{H}_P = \sum_{i=1}^{N} h_i \sigma_i^z + \sum_{i,j=1}^{N} J_{ij} \sigma_i^z \sigma_j^z$$



Envelope functions



Chimera(3,3,4) graph

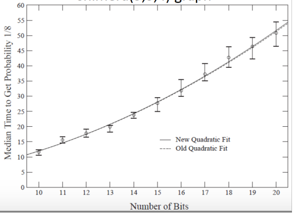
Actual DW-1 has Chimera(4,4,4) layout: 128 qubits

Applications:

Ising $\operatorname{argmin}_{s} \left\{ \langle s, Js \rangle + \langle h, s \rangle \right\}$ **NP-hard**

QUBO $\operatorname{argmin}_{x} \langle x, Qx \rangle$

Significant quantum speedup in time complexity is expected but not quantified theoretically



E. Farhi et al "A Quantum Adiabatic Evolution Algorithm Applied to Random Instances of an NP-Complete Problem"10.1126/science.1057726"

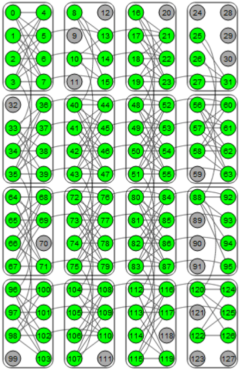
DW-1: programming the chip

$$\mathcal{H}_{\mathrm{P}} = \sum_{i=1}^{N} h_i \sigma_i^z + \sum_{i,j=1}^{N} J_{ij} \sigma_i^z \sigma_j^z$$

Outline:

1. Assign ‘h’ and ‘J’ values;
2. Call the solver to implement the quantum annealing process. Parameters:

   a. Annealing time (1000 – 20000 μs)
   b. Number of measurements
   c. Thermalization time
3. Output: Measurement outcomes (0/1 bit strings for QUBO and -1/1 for Ising) and their probabilities

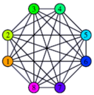


Real-time connectivity graph

DW-1: Hardware Implementation Issues

Ising $\underset{s}{\operatorname{argmin}}\left\{\langle \boldsymbol{s}, \boldsymbol{J}\boldsymbol{s}\rangle + \langle \boldsymbol{h}, \boldsymbol{s}\rangle\right\}$

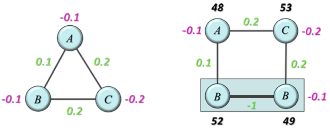
QUBO $\underset{x}{\operatorname{argmin}}\langle \boldsymbol{x}, \boldsymbol{Q}\boldsymbol{x}\rangle$



Issue #1 - Connectivity:

May be different from what the problem requires

Solution: Embedding



Desired **Actually implemented**

Drawbacks: chip-specific, hard to keep identical states for long spin chains.

Issue #2 – Precision:

Each ‘h’ and ‘J’ can be encoded with only 3-bit precision

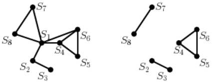
Solution: Splitting

$$h_i s_i \rightarrow (h_i/3)(q_i^1 + q_i^2 + q_i^3)$$

Issue #3 – Qubit number:

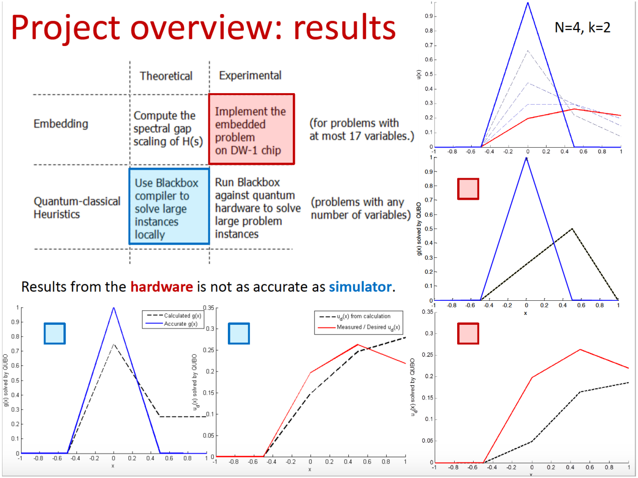
Current chip at ISI supports up to **17** fully-connected qubits embedding.

Solution: Classical heuristics + QA



Cut-set conditioning

Operating D-Wave One



Is D-Wave really a quantum annealer?

Quantum annealing with more than one hundred qubits

Sergio Boixo, Troels F. Rønnow, Sergei V. Isakov, Zhihui Wang, David Wecker, Daniel A. Lidar, John M. Martinis, Matthias Troyer

(Submitted on 16 Apr 2013 (v1), last revised 21 Jul 2013 (this version, v2))

Quantum technology is maturing to the point where quantum devices, such as quantum communication systems, quantum random number generators and quantum simulators, may be built with capabilities exceeding classical computers. A quantum annealer, in particular, solves hard optimisation problems by evolving a known initial configuration at non-zero temperature towards the ground state of a Hamiltonian encoding a given problem. Here, we present results from experiments on a 108 qubit D-Wave One device based on superconducting flux qubits. The strong correlations between the device and a simulated quantum annealer, in contrast with weak correlations between the device and classical annealing or classical spin dynamics, demonstrate that the device performs quantum annealing. We find additional evidence for quantum annealing in the form of small-gap avoided level crossings characterizing the hard problems. To assess the computational power of the device we compare it to optimised classical algorithms.

Classical signature of quantum annealing

John A. Smolin, Graeme Smith

(Submitted on 21 May 2013)

A pair of recent articles concluded that the D-Wave One machine actually operates in the quantum regime, rather than performing some classical evolution. Here we give a classical model that leads to the same behaviors used in those works to infer quantum effects. Thus, the evidence presented does not demonstrate the presence of quantum effects.

Comment on: "Classical signature of quantum annealing"

Lei Wang, Troels F. Rønnow, Sergio Boixo, Sergei V. Isakov, Zhihui Wang, David Wecker, Daniel A. Lidar, John M. Martinis, Matthias Troyer

(Submitted on 24 May 2013)

In a recent preprint (arXiv:1305.4904) entitled "Classical signature of quantum annealing" Smolin and Smith point out that a bimodal distribution presented in (arXiv:1304.4595) for the success probability in the D-Wave device does not in itself provide sufficient evidence for quantum annealing, by presenting a classical model that also exhibits bimodality. Here we analyze their model and in addition present a similar model derived from the semi-classical limit of quantum spin dynamics, which also exhibits a bimodal distribution. We find that in both cases the correlations between the success probabilities of these classical models and the D-Wave device are weak compared to the correlations between a simulated quantum annealer and the D-Wave device. Indeed, the evidence for quantum annealing presented in arXiv:1304.4595 is not limited to the bimodality, but relies in addition on the success probability correlations between the D-Wave device and the simulated quantum annealer. The Smolin-Smith model and our semi-classical spin model both fail this correlation test.

IBM Cloud Quantum Computing

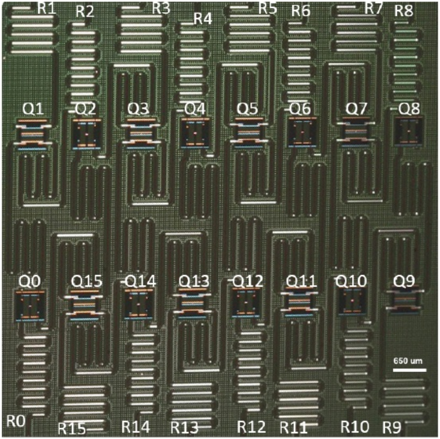
IBM Quantum Computing Learn Login

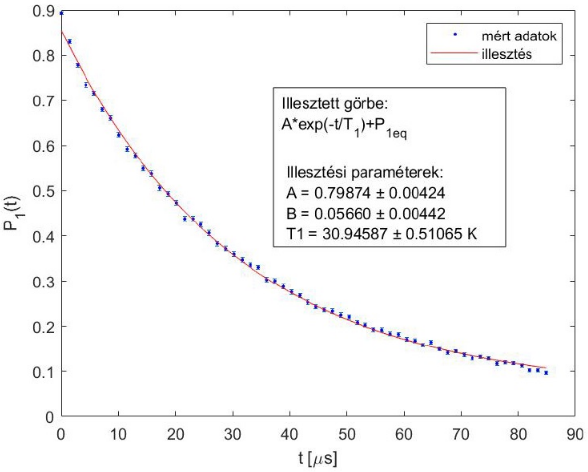
Explore the IBM Quantum Experience

The IBM Quantum Experience is a virtual lab where you can design and run your own algorithms through the cloud on real quantum processors located in the IBM Quantum Lab at the Thomas J Watson Research Center in Yorktown Heights, NY.

Log-in Sign up

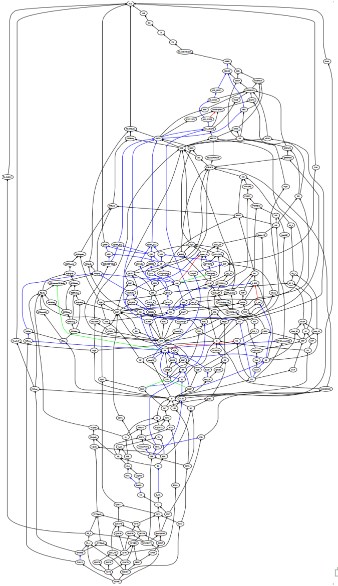
IBM Cloud Quantum Computing



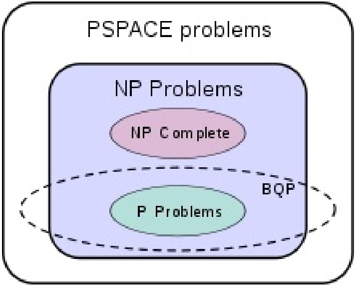


Taken from Ákos Budai's BSc Thesis.

Computational Complexity

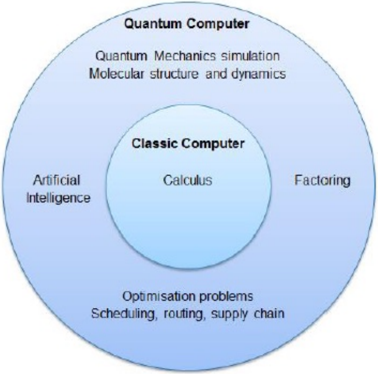


Computational Complexity



What are the future perspectives?

Quantum computing has a much larger reach than a classic computer – and thus a much larger potential addressable market, in our view



Source: Morgan Stanley Research

Quantum Computing and GPUs

Simulating Quantum Computers Using OpenCL

Adam Kelly

May 1, 2018

I present QCGPU, an open source Rust library for simulating quantum computers. QCGPU uses the OpenCL framework to enable acceleration by devices such as GPUs, FPGAs and DSPs. I perform a number of optimizations including parallelizing operations such as the application of gates and the calculation of various state probabilities for the purpose of measurment. Using an Amazon EC2 p3.2xLarge instance, the library is then benchmarked and also compared against some preexisting libraries with the same purpose. The presented library is limited only by the memory of the host machine or that of the device being used by OpenCL. The finished software is available at https://github.com/qcgpu/qcgpu-rust.

1 Introduction

Quantum computers are thought to be the key to some types of problems, such as factoring a semi-prime integer [4] [17], calculating discrete logarithms, the search for an element in an unstructured database [9] [22], super dense coding [21], simulation of quantum systems, along with many other algorithms. Currently, the Quantum Algorithm Zoo, a website that details many algorithms for quantum computers cites 386 papers, at the time of writing [19]. It has also been suggested that quantum computers could create new opportunities in the fields of chemistry [12], optimization [14] and machine learning [16].

While it is not feasible to solve some of these problems on classical computers, the quantum algorithms do not violate the Church-Turing theorem and thus can be, to a small extent, simulated using classical computers.

There are some real quantum computers, such as IBM's quantum experience [6], which has semi-public access to a 5-qubit machine, a 16-qubit machine and a 20-qubit machine through their software library *qiskit* [1]. With devices now providing up to 20 controllable qubits, there are many issues being raised, including (most importantly) the ability to assess the correctness, performance and scalability of quantum algorithms.

It is this issue which simulators of quantum computers address. They allow the user to test quantum algorithms using a limited number of qubits and calculate measurements, state amplitudes and occasionally implement features which help in this testing process such as density matrices.

2 Background

2.1 Existing Research

There are many existing quantum computer simulators (many are listed at [2]) along with some existing proposals for GPU accelerated simulators. These include simulations using a large number of qubits and memory [11], using proprietary frameworks such as CUDA [3] [10].

To the author's knowledge, QCGPU is the first open source quantum computer simulator to use the functionality provided by OpenCL. The advantages/disadvantages of which (over CUDA or similar frameworks) are discussed in section 2.2.

2.2 OpenCL

OpenCL (Open Computing Language) is a general-purpose framework for heterogeneous parallel computing on cross-vendor hardware, such as CPUs, GPUs, DSP (digital signal processors) and FPGAs (field-programmable gate arrays). It provides an abstraction for low-level hardware routing and a consistent memory and execution model for dealing with massively-parallel code execution. This allows the framework to scale from embedded systems to hardware from NVidia, API, AMD, Intel and other manufacturers, all without having to rewrite the source code for various backends. An overview of OpenCL is given in [20].

The main advantage of using OpenCL over a hardware specific framework is that of a portability first approach. OpenCL has the largest hardware coverage, and as a library only it requires no tool dependencies. Aside from this, OpenCL is very well suited to tasks that can be expressed as a program working in parallel over simple data structures (such as arrays/vectors). The disadvantages with OpenCL, however, come from this lack of a hardware-

Adam Kelly: adamkelly2201@gmail.com