ZOLTÁN ZIMBORÁS

A GENTLE INTRODUCTION TO QUANTUM COMPUTING

WIGNER HUNQUTECH QHUNGARY

ZOLTÁN ZIMBORÁS

FIRST AN INTRODUCTION TO OUR WORKSHOP

WIGNER HUNQUTECH Q HUNGARY

Why did we organized this workshop now?

Why did we organized this workshop now?

NEWS AND VIEWS · 23 OCTOBER 2019

Quantum computing takes flight

A programmable quantum computer has been reported to outperform the most powerful conventional computers in a specific task — a milestone in computing comparable in importance to the Wright brothers' first flights.

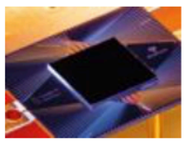
William D. Oliver

Quantum computers promise to perform certain tasks much faster than ordinary (classical) computers. In essence, a quantum computer carefully orchestrates quantum effects (superposition, entanglement and interference) to explore a huge computational space and ultimately converge on a solution, or solutions, to a problem. If the numbers of quantum bits (qubits) and operations reach even modest levels, carrying out the same task on a state-of-the-art supercomputer becomes intractable on any reasonable timescale — a regime termed quantum computational supremacy[1]. However, reaching this regime requires a robust quantum processor, because each additional imperfect operation incessantly chips away at overall performance. It has therefore been questioned whether a sufficiently large quantum computer could ever be controlled in practice. But now, in a paper in *Nature*, Arute *et al.*[2] report quantum supremacy using a 53-qubit processor.

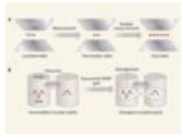
PDF version

RELATED ARTICLES

Read the paper: Quantum supremacy using a programmable superconducting processor



Promising ways to encode and manipulate quantum information



A milestone in quantum computing

Why did we organized this workshop now?

Software & Consultants

Quantum Consultants
1QBit
| Entanglement Partners_ >
Zapata Computing
CQC Cambridge Quantum Computing Limited
Atos
QXBranch
Entropica Labs
Quantum computing, at your service
quantum benchmark
Q-CTRL
ProteinQure
NETRAMARK
Qu&Co
Artiste-qb.net
QC WARE
Everettian
BRA-KET SCIENCE
Quantika
Quantum deep tech
EeroQ QILIMANJARO
Strange Works
⟨Qubit|Era⟩
Horizon Quantum Computing
River Lane Research

Quantum Computers

IBM
D-Wave
The Quantum Computing Company™
Google
Microsoft
rigetti
Silicon Quantum Computing
IONQ
Alibaba Group
NTT
XANADU
qci
Bell Labs
NOKIA Bell Labs
EeroQ QILIMANJARO
Alpine Quantum Technologies
Oxford Quantum Circuits

Enabling Technologies

Qubitekk
BlueFors
Leiden Cryogenics
Leader in Low Temperature Techniques
Oxford Cryosystems
Zurich Instruments
ANYON
Computing with Manufactured Quanta™
intel
KEYSIGHT TECHNOLOGIES
Signadyne
Metempsy
Raytheon BBN Technologies
Quandela
HYPRES
The Digital Superconductor Company™
JANIS
Q-CTRL
SPICE LABS
Delft Circuits
quantum computing hardware

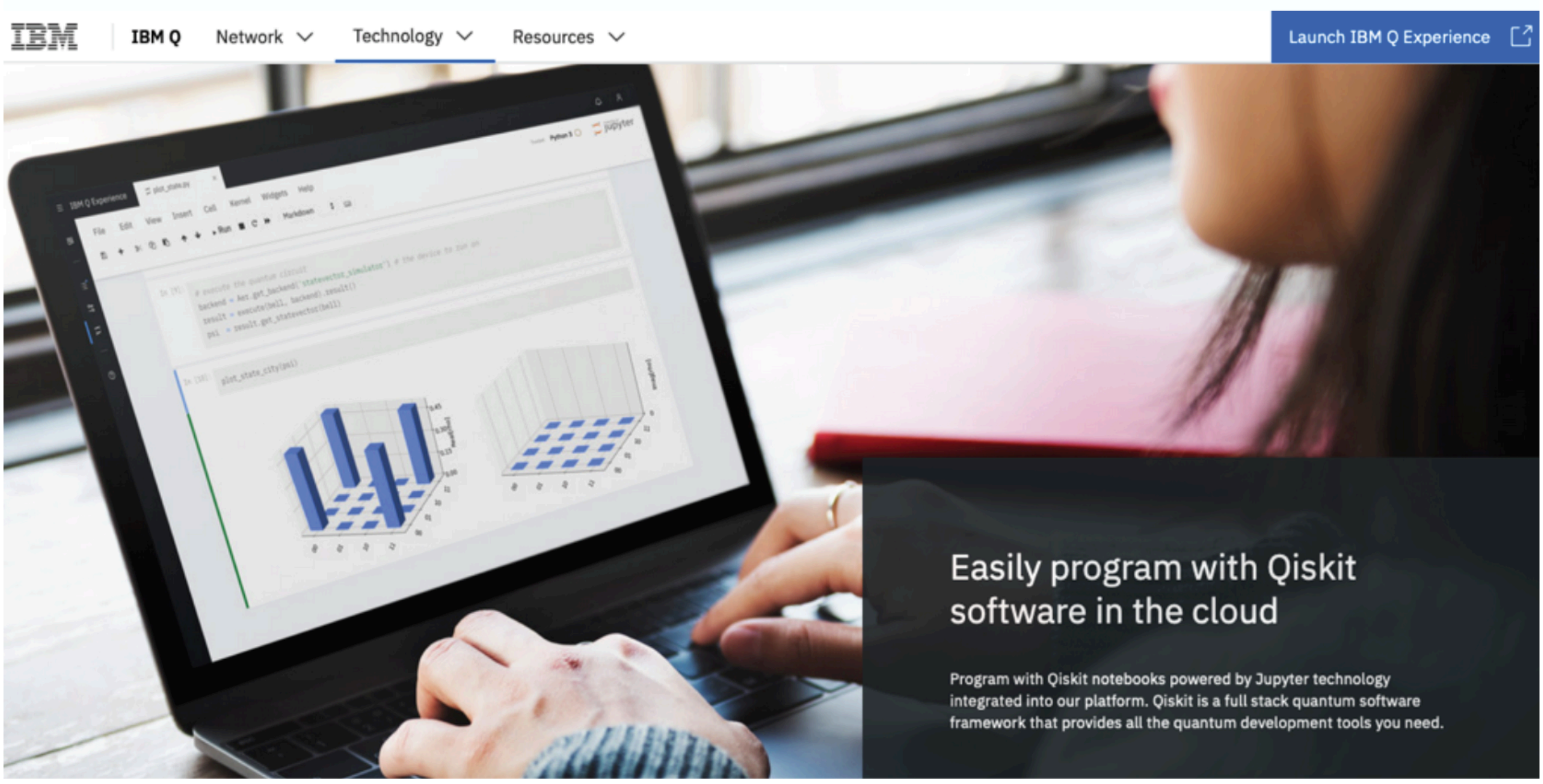
New Funding Strategies

Qubit Protocol
QuantumX
Creative Destruction Lab
The Quantum Revolution Fund
Quantum Valley Investments

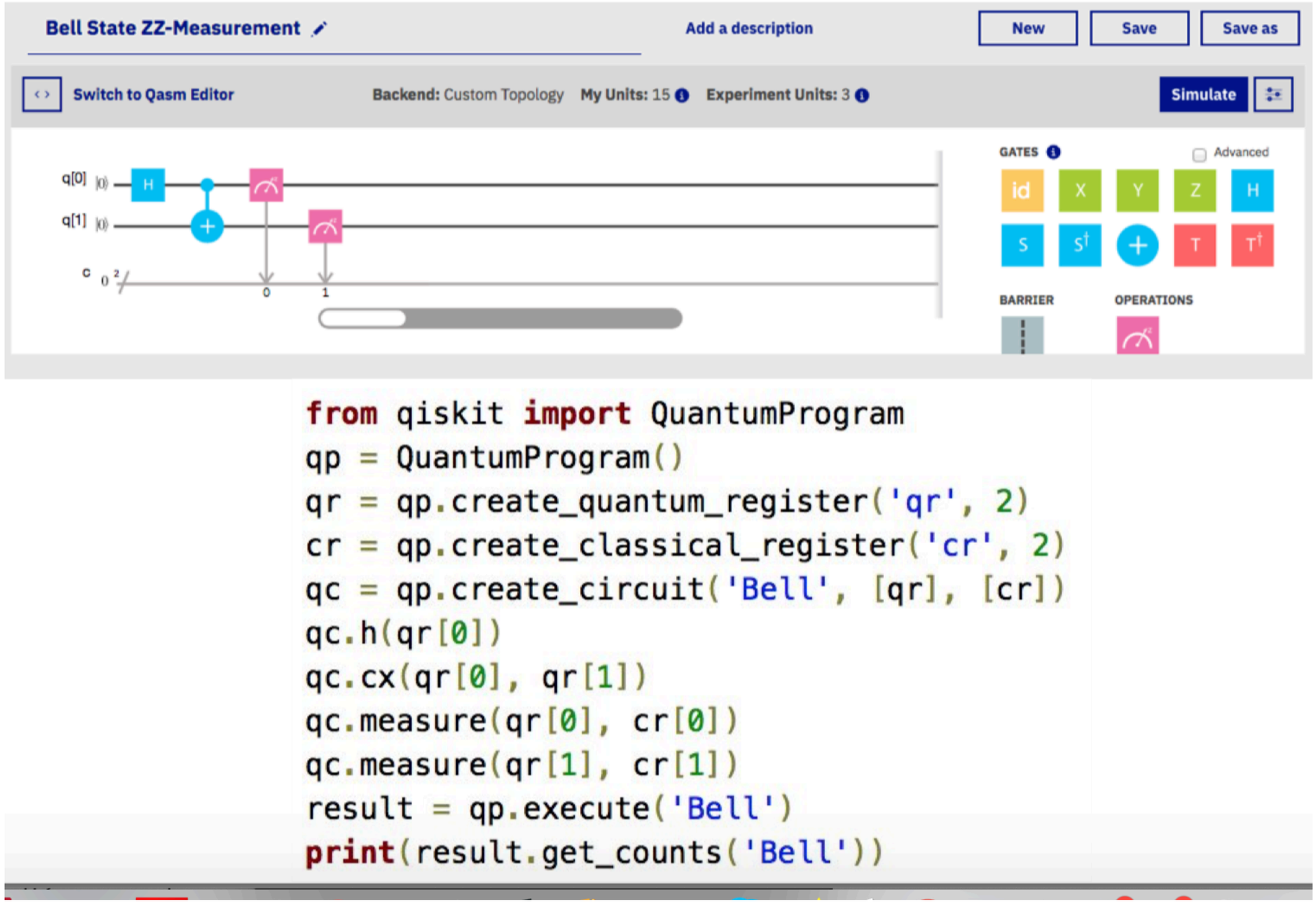
Representative list of players. A very active ecosystem!

Quantum World Association

Why did we organized this workshop now?



There is a handy python environment also for quantum programming the IBM machines




```
from qiskit import QuantumProgram
qp = QuantumProgram()
qr = qp.create_quantum_register('qr', 2)
cr = qp.create_classical_register('cr', 2)
qc = qp.create_circuit('Bell', [qr], [cr])
qc.h(qr[0])
qc.cx(qr[0], qr[1])
qc.measure(qr[0], cr[0])
qc.measure(qr[1], cr[1])
result = qp.execute('Bell')
print(result.get_counts('Bell'))
```

Mitigation of the readout noise

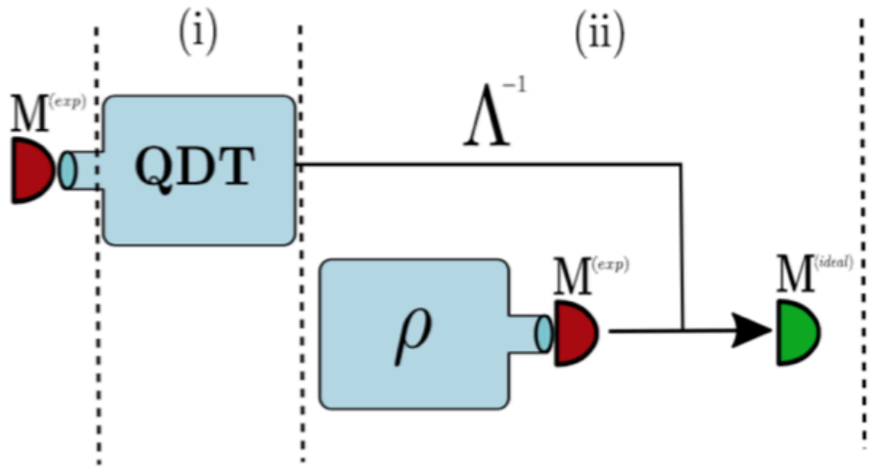


FIG. 1: Pictorial representation of a correction procedure.

Mitigation of readout noise by classical post-processing based on Quantum Detector Tomography

Filip B. Maciejewski,[1,*] Zoltán Zimborás,[2,3,†] and Michał Oszmaniec[4,‡]

We propose a simple scheme to reduce readout errors in experiments on quantum systems with finite number of measurement outcomes. Our method relies on performing classical post-processing which is preceded by Quantum Detector Tomography, i.e., the reconstruction of a Positive-Operator Valued Measure (POVM) describing the given quantum measurement device. If the reconstructed POVM differs from the ideal one only by an invertible classical noise, it is possible to correct the outcome statistics of other experiments performed on the same device. We provide empirical arguments that a classical noise might be the dominant form of measurement noise in contemporary quantum devices consisting of transmon qubits. We also analyze the influence of both finite-size statistics and non-classical errors on the performance of our correction scheme. Finally, we provide a characterization of readout noise occurring in IBM quantum devices and test our mitigation scheme on these. We observe a large improvement of results for a number of tasks including Quantum State Tomography (QST), Quantum Process Tomography (QPT), implementation of non-projective measurements, implementation of a certain probability distributions and implementation of certain quantum algorithm's - Grover's search and the Bernstein-Vazirani algorithm.

Why did we organized this workshop now?

AQT

HOME ABOUT TECHNOLOGY NEWS & EVENTS CAREER CONTACT

AQT QUANTUM COMPUTER LEVERAGES CIRQ FOR ALGORITHM DEVELOPMENT

September 2019: Quantum computers promise to solve problems that are out of reach for today's supercomputers. Programming quantum computers differs radically from what programmers are used today and thus new programming languages are required. A collaborative effort by Alpine Quantum Technologies (AQT) and the University of Innsbruck allows direct access to the ion-trap quantum computer in Innsbruck via Cirq, a framework developed by Google focused on developing and implementing quantum algorithms. Cirq can be used to explore quantum algorithms on the different hardware architectures, superconducting electronics and trapped ions.

For more details, please check out this press release of the University of Innsbruck.

AQT co-founders Prof. Peter Zoller (left) and Dr. Thomas Monz (right) with Google's Dr. Markus Hoffmann (centre). Photo: M. R. Knabl

Why did we organized this workshop now?

Quantum Computing

Qiskit – Write once, target multiple architectures

November 5, 2019 | Written by: Ali Javadi-Abhari, Paul Nation, and Jay Gambetta

Categorized: Quantum Computing

Share this post:

Ever since we released the first superconducting quantum processor online as a hosted device over three years ago, we have witnessed an explosion of engagement and new research generated through that access. Yet, the community is broader still, with a diverse set of experimental platforms and system builds quite different in various aspects to our own superconducting qubit devices. Hence, to truly accelerate research and development, we need a software framework that can be universally applied across the available qubit technologies, with the overarching goal of enabling novel quantum experimental demonstrations. Creating an open software platform supporting this broad effort is the only way we will truly succeed with quantum computing.

From its outset, our open-source Qiskit quantum computing framework has been designed to be extensible, and to support research beyond the IBM quantum systems based on superconducting qubits. This inherent flexibility allows for straightforward integration of additional gates, optimization passes, and providers (sources of different systems). Taken together, Qiskit has the flexibility to target different underlying quantum hardware with minimal additions to its code base. To demonstrate this, we have recently added support in Qiskit for trapped ion-based quantum computing devices, and enabled access to the five-qubit trapped ion device at the University of Innsbruck (UIBK) hosted by Alpine Quantum Technologies (AQT). This device resides in Innsbruck Austria, nearly 6,500 km away from where the IBM quantum systems in New York.

Why did we organized this workshop now?

On Wed, Nov 13, 2019 at 2:35 AM XXXX wrote:

---------- Forwarded message ---------
From: **ZZZZ**
Date: Tue, 12 Nov 2019, 16:28
Subject: XXXX
To: XXXX, YYYY

Hi XXXX and XXXX,

XXXX, I first owe you an apology for my incredibly delayed reply to your LinkedIn message. The past few months have been hectic for me while I transition to a new team, though that is no excuse for leaving our conversation hanging.

That's where my colleague YYYY comes in. I'd like to introduce you two as YYYY directly works with our research partners and users of our platform. XXXX, you mentioned in your message to me that our XXXX platform did not perform as well and that you were waiting to hear back about QPU access. Amy is the person to speak to regarding feedback, access, and more.

YYYY, for reference here is a paper XXXX and his collaborators released in July on NISQ readout noise that prompted our conversation. He and his team are based in Warsaw and I'd love to find a way for us to support their research efforts, especially as they continue to focus their work more on quantum computation.

I'll let you two take it from here, though please let me know if there is any way I can be of further assistance.

All the best,
ZZZZ

--

Send

QWORLD AND QHUNGARY

QWORLD

QCOUSINS QWOMEN QJUNIOR QKITCHEN QUNIVERSITY EVENTS PROJECTS QWORL

WELCOME TO QWORLD!

QWorld was established by five quantum cousins during QDrive project in July 2019. At the moment, QWorld has six channels for working on different target groups and people:

QCousins, QWomen, QJunior, QKitchen, QUniversity, and QMentor Training.

Our main aim is to have an open access and public global ecosystem for quantum technologies and quantum software by the year 2025 so that each interested hardworking individual, group, institute, or region can be easily part of the ecosystem.

We are looking for enthusiastic individuals, groups, institutions, organization, and companies to take part in QWorld or to work and operate together.

Join us/Invite us/Collaborate with us.

QWORLD AND QHUNGARY

- What is a quantum computer, what can it do, how does it work?
- How can I write my own programs and run them on **IBM's quantum computer**?

If you wonder about these questions, have had some experience with linear algebra, a touch of quantum physics, and basic familiarity with python, workshops of QHungary are for you!

QHungary is a group of people enthusiastic about, and with experience in, quantum computing. The founding members are researchers at the Wigner Research Centre for Physics of the Hungarian Academy of Sciences, the Budapest University of Technology and Economics, and Eötvös Loránd University. Our aim is to bring the fun and fascination of quantum computing to a wider audience, including high school students and people from outside academia, through hands-on workshops.

QHungary is part of **QWorld**, an international network of quantum programming instruction groups. QWorld was born out of a project from **QLatvia**: **QDrive,** a travelling quantum programming workshop of summer 2019 - reaching 400 participants, by driving 20.000 km through 20 countries in 100 days**. QDrive Budapest** introduced 50 participants to quantum programming using python and **qiskit**, through workshops instructed by QLatvia. The material is a series of ***Jupyter notebooks***, using the qiskit package, with few tasks each.

QHungary Team

Founding Members:

Zoltán Zimborás (Wigner RCP), János Asbóth (BME), László Oroszlány (Eötvös Univ), András Pályi (BME)

Contact:

zimboras.zoltan@wigner.mta.hu

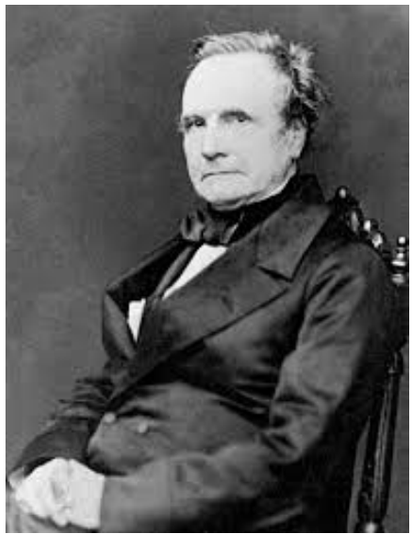
Next workshop: November 21-22, 2019

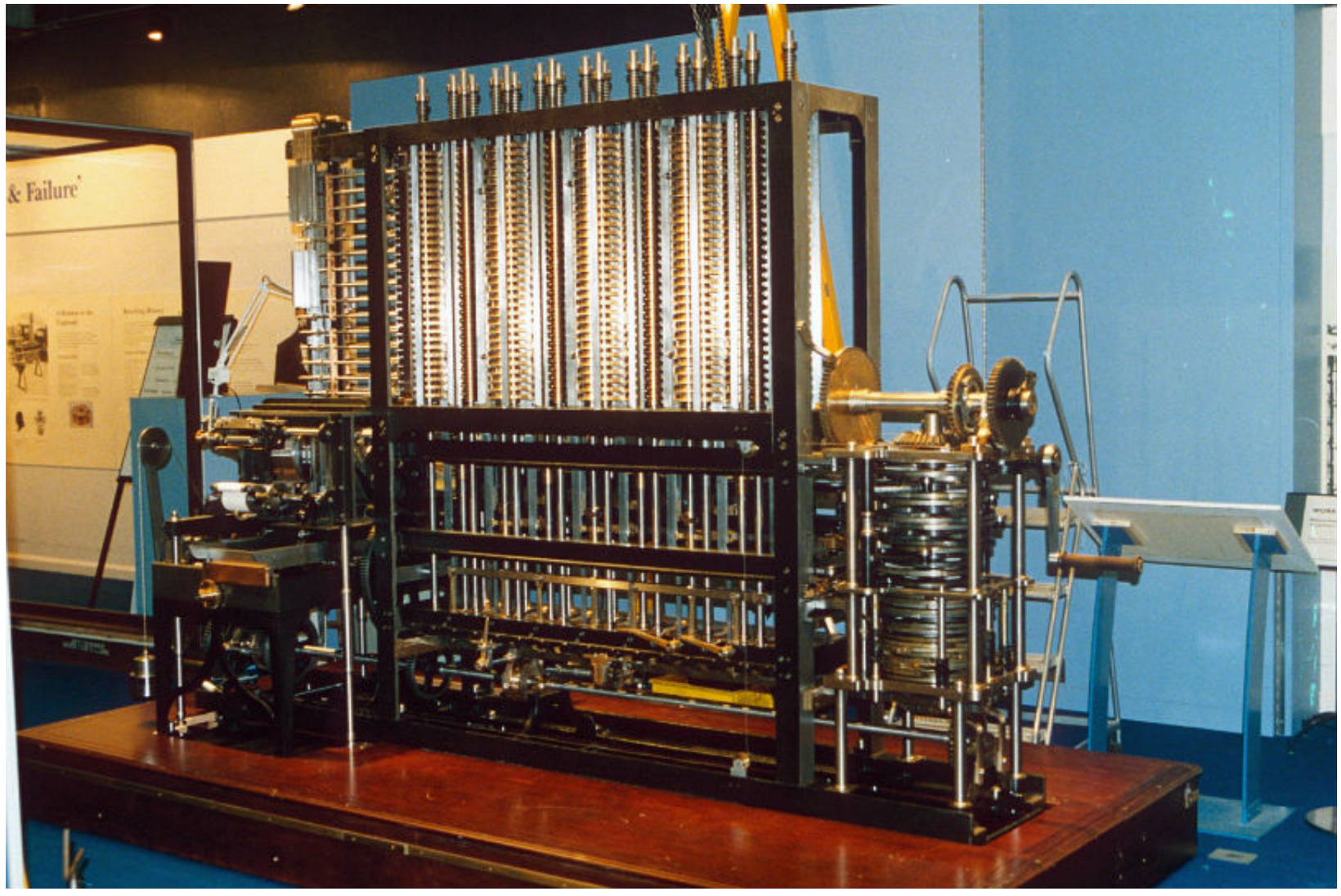
ZOLTÁN ZIMBORÁS

A GENTLE INTRODUCTION TO FROM CLASSICAL TO QUANTUM COMPUTING

WIGNER HUNQUTECH QHUNGARY

Early approaches to computing machines and programs

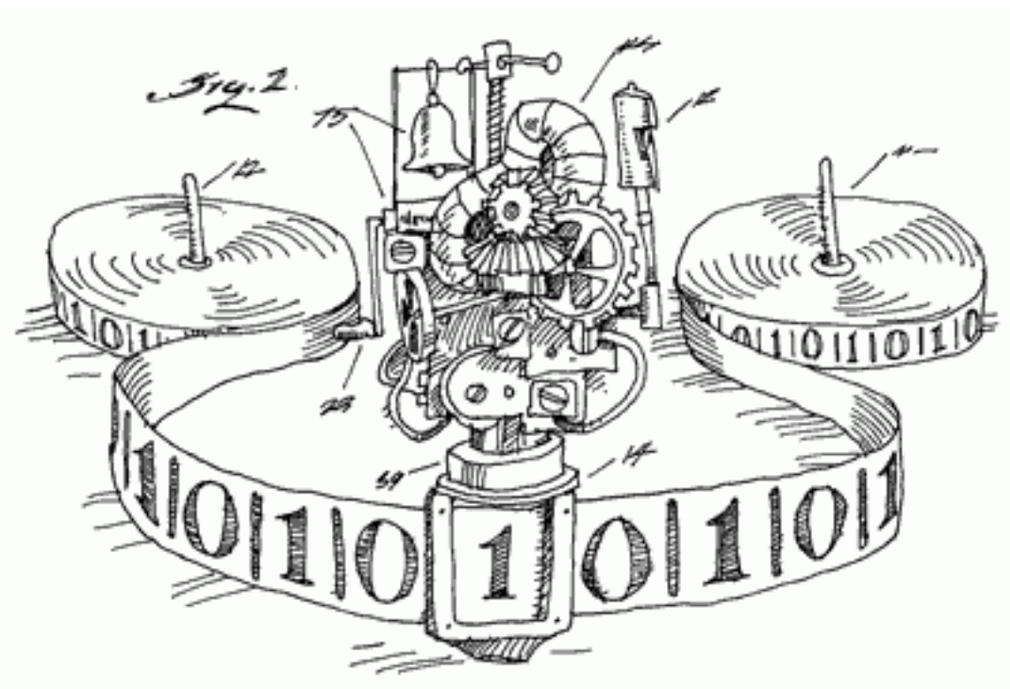


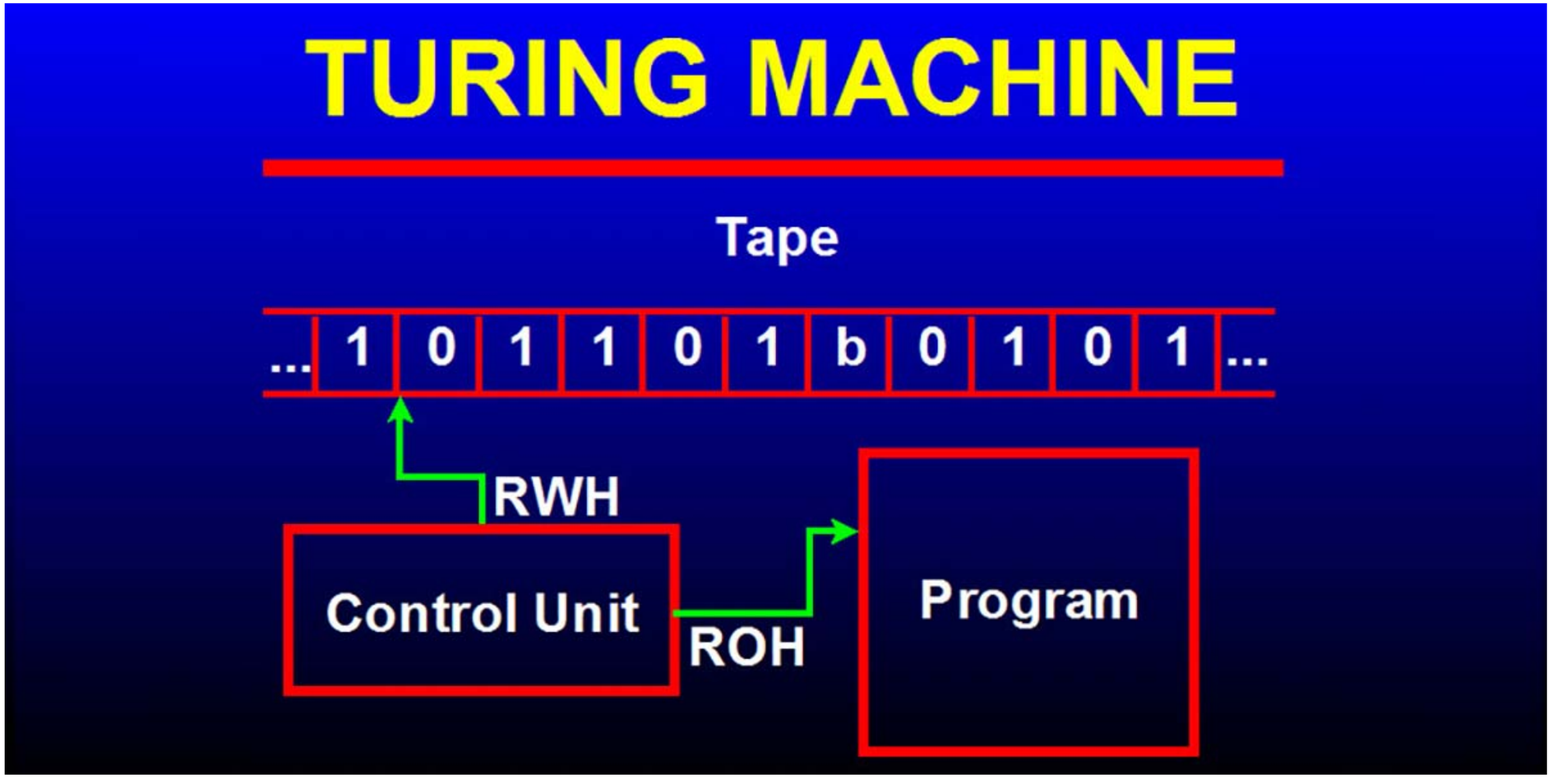


Charles Babbage's Difference and Analytical Engines

Ada Lovelace's program to calculate Bernoulli numbers

Definition of Computation and the Turing Machine





The extended Church-Turing thesis

Church-Turing thesis (simple version):
Everything that is computable is computable by a Turing machine

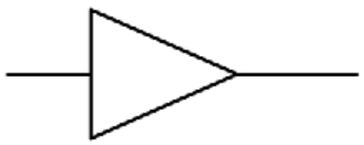
The extended Church-Turing thesis

Church-Turing thesis (simple version):
Everything that is computable is computable by a Turing machine

The extended Church-Turing thesis:
Any "reasonable" model of computation can be *efficiently* simulated on a probabilistic Turing machine (an efficient simulation is one whose running time is bounded by some polynomial in the running time of the simulated machine).

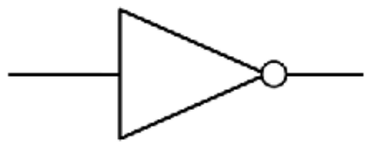
Bits and Boolean Gates

YES gate



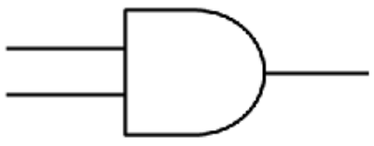
INPUT	OUTPUT
0	0
1	1

NOT gate



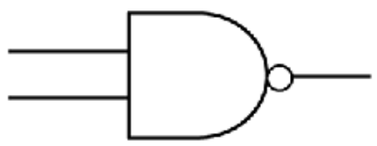
INPUT	OUTPUT
0	1
1	0

AND gate



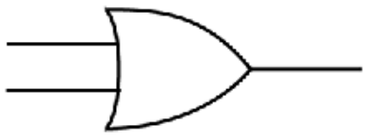
INPUT		OUTPUT
A	B	
0	0	0
0	1	0
1	0	0
1	1	1

NAND gate



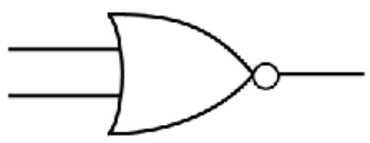
INPUT		OUTPUT
A	B	
0	0	1
0	1	1
1	0	1
1	1	0

OR gate



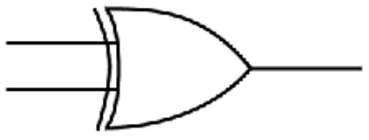
INPUT		OUTPUT
A	B	
0	0	0
0	1	1
1	0	1
1	1	1

NOR gate



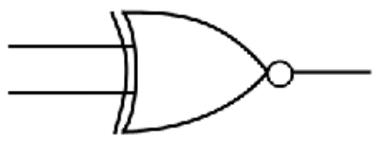
INPUT		OUTPUT
A	B	
0	0	1
0	1	0
1	0	0
1	1	0

XOR gate



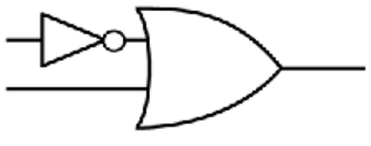
INPUT		OUTPUT
A	B	
0	0	0
0	1	1
1	0	1
1	1	0

XNOR gate



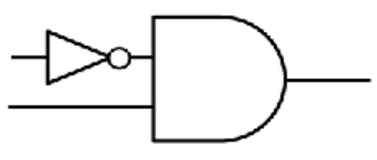
INPUT		OUTPUT
A	B	
0	0	1
0	1	0
1	0	0
1	1	1

IMPLY
(A IMPLY B)



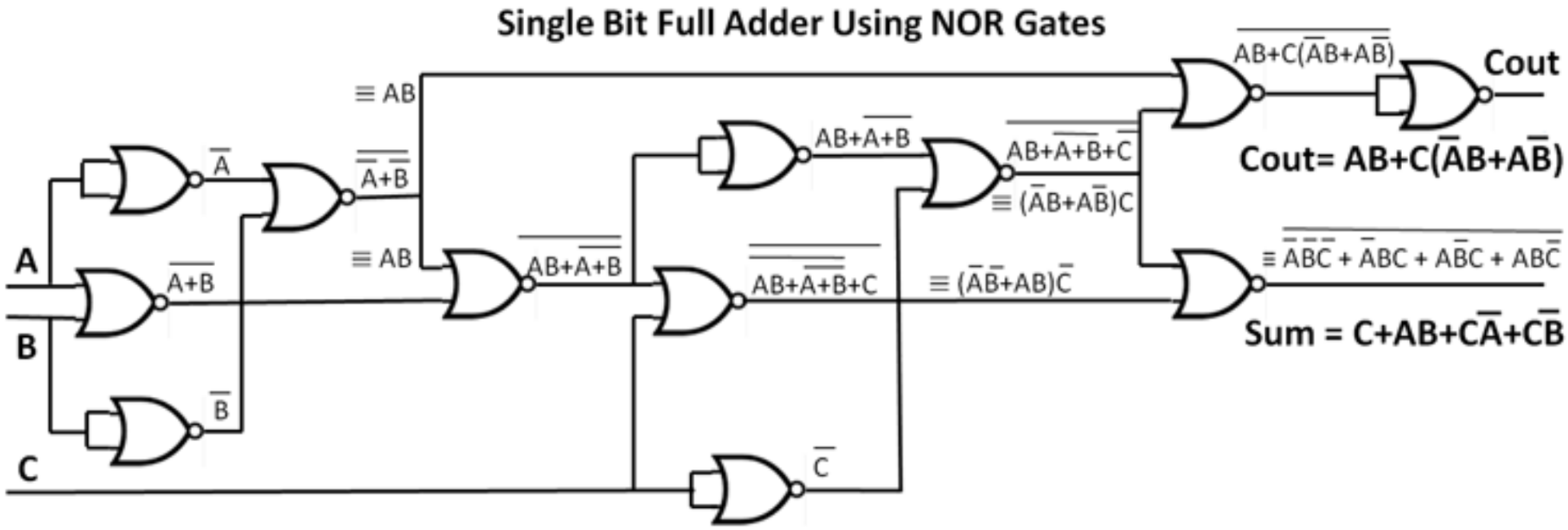
INPUT		OUTPUT
A	B	
0	0	1
0	1	1
1	0	0
1	1	1

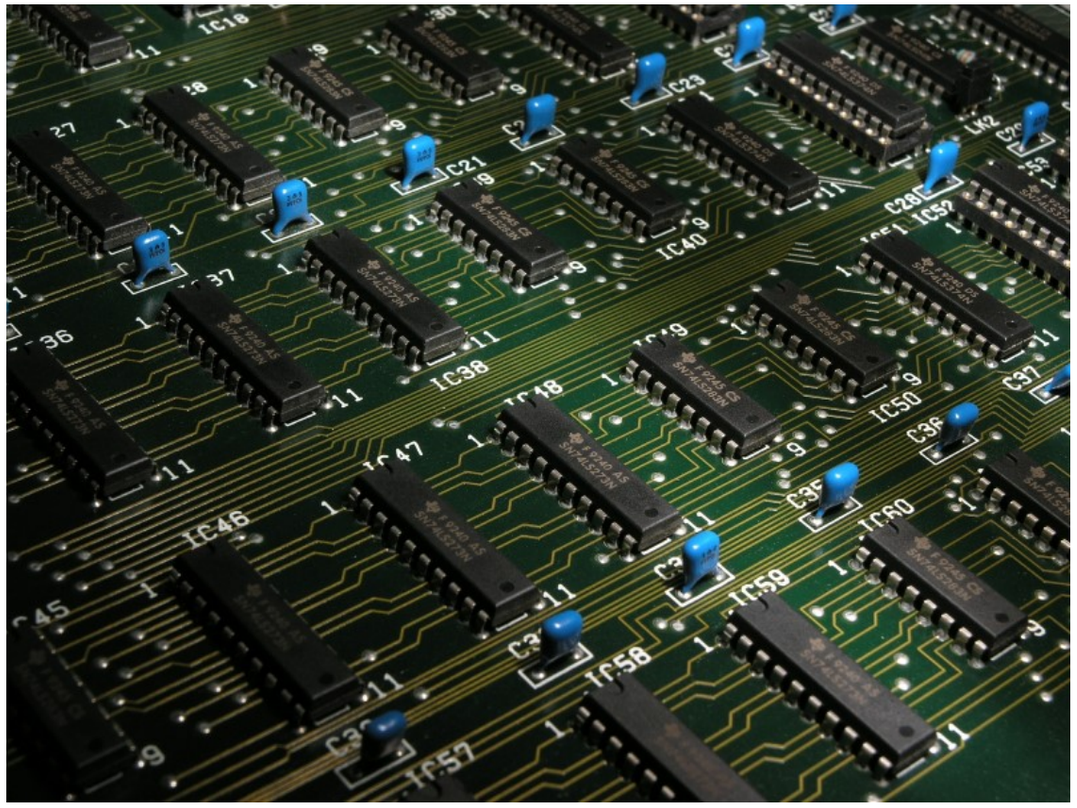
N-IMPLY
(B N-IMPLY A)



INPUT		OUTPUT
A	B	
0	0	0
0	1	1
1	0	0
1	1	0

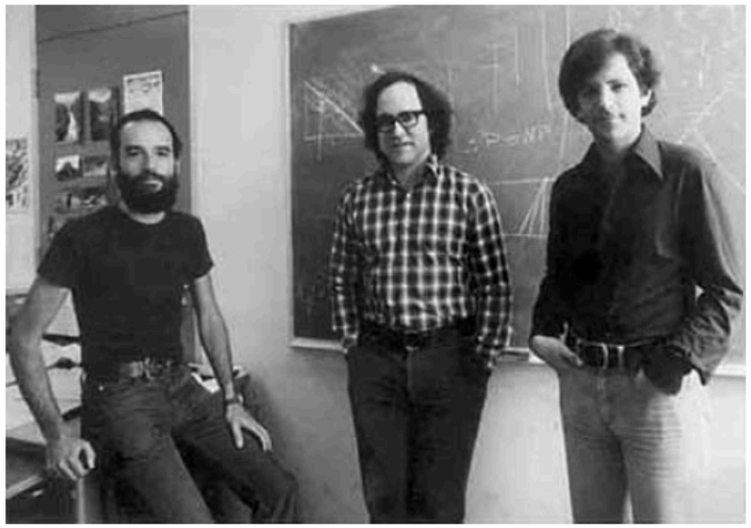
Circuits of Gates





Related problems can have very different complexity

- 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 = 123018668453011775513049495838496272077285356959533479219732245215172640050726365751874520219978646938995647494277406384592519255732630345373154826850791702612214291346167042921431160222124047927473779408066535141959745985690214341 3
```

```
RSA-768 = 33478071698956898786044169848212690817704794983713768568912431388982883793878002287614711652531743087737814467999489
        × 36746043666799590428244633799627952632279158164343087642676032283815739666511279233373417143396810270092798736308917
```

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 5 years, 11 months ago
viewed 27,215 times
active 2 years, 11 months ago

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

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:

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

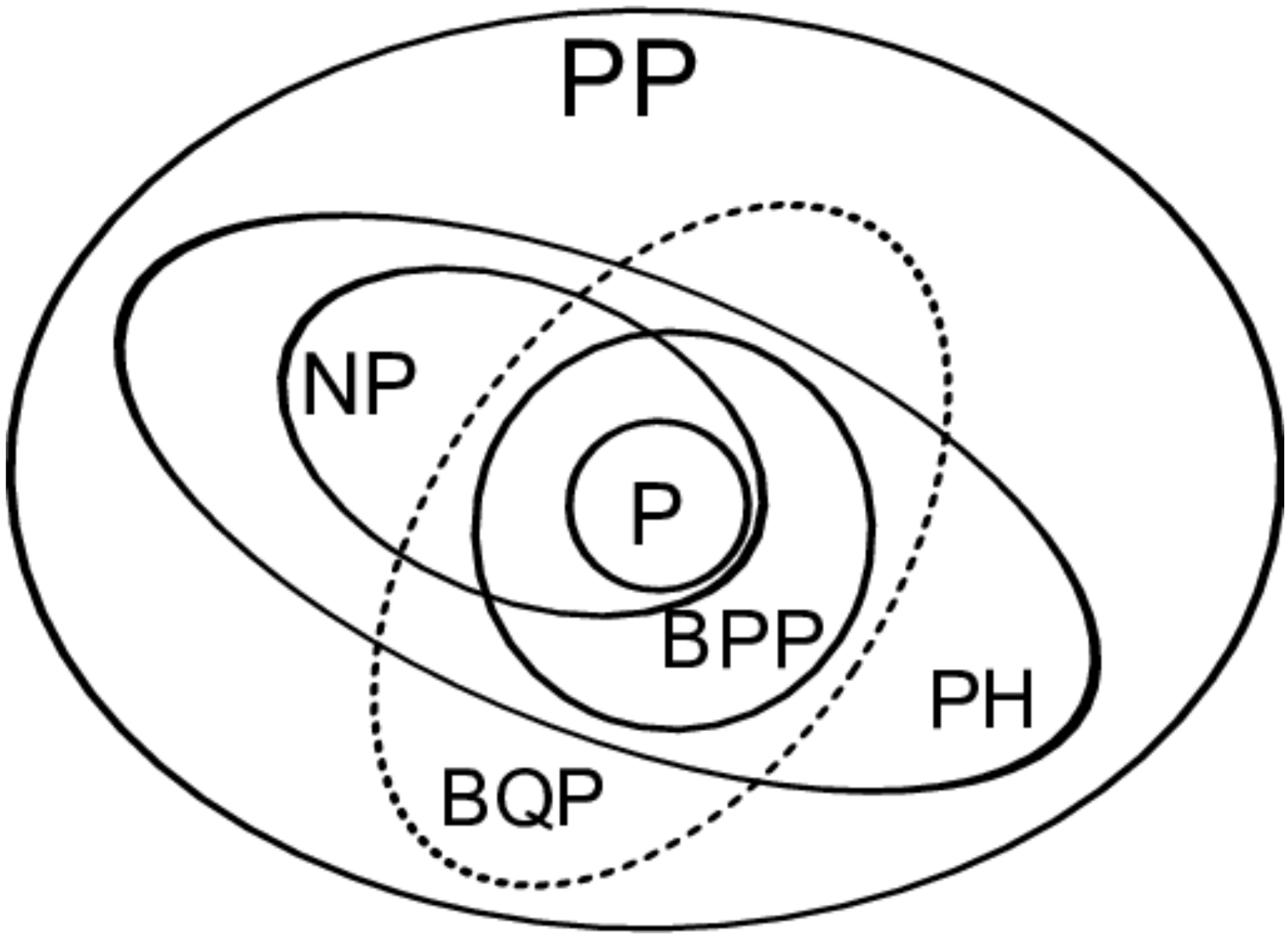
The number of RSA moduli (i.e. pair of two distinct primes) is therefore:

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

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:

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

Complexity classes



The extended Church-Turing thesis

Church-Turing thesis (simple version):
Everything that is computable is computable by a Turing machine

The extended Church-Turing thesis:
Any "reasonable" model of computation can be *efficiently* simulated on a probabilistic Turing machine (an efficient simulation is one whose running time is bounded by some polynomial in the running time of the simulated machine).

Feynman's question and vision

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.

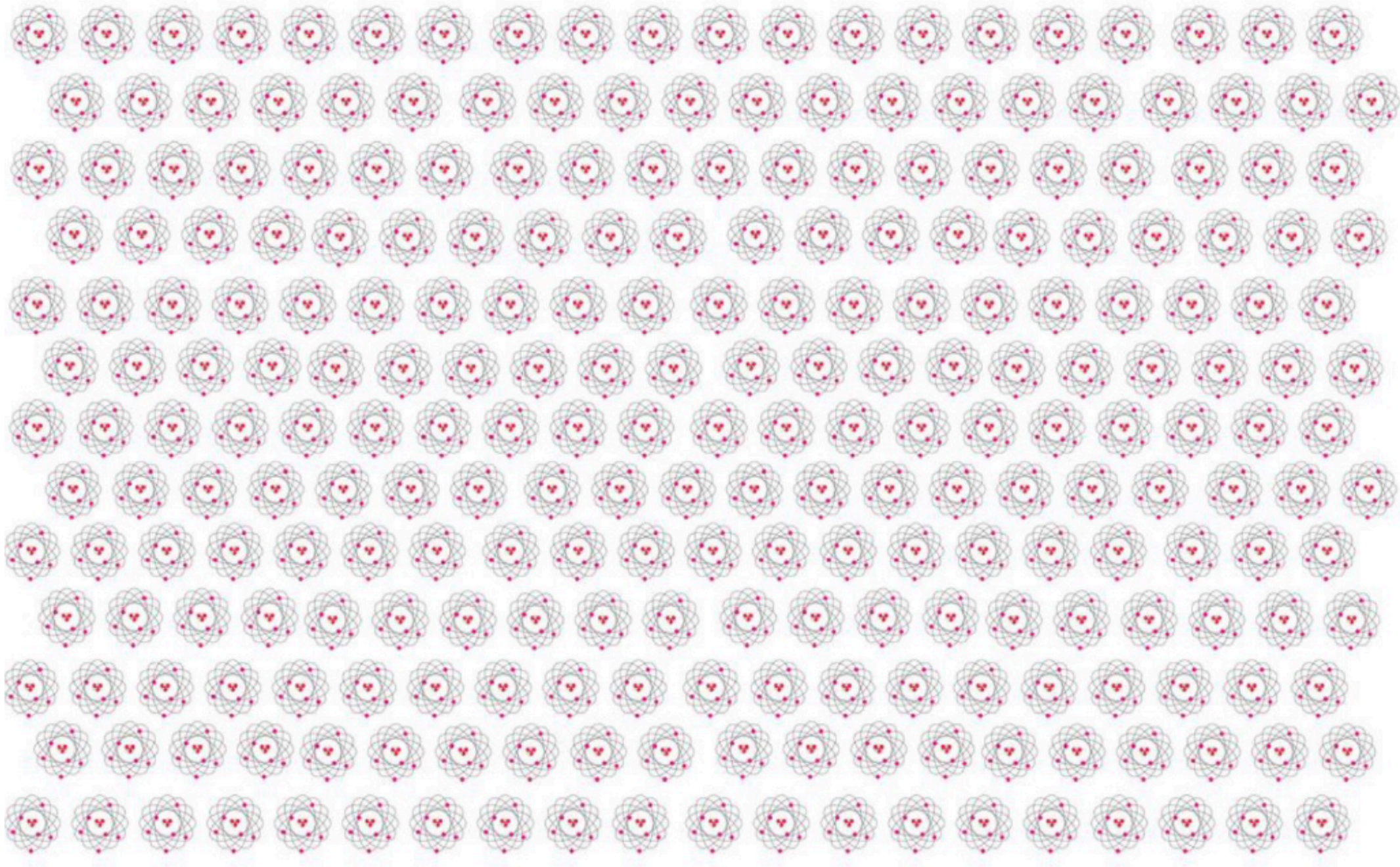
The first question is, What kind of computer are we going to use to simulate physics? Computer theory has been developed to a point where it realizes that it doesn't make any difference; when you get to a *universal computer*, it doesn't matter how it's manufactured, how it's actually made. Therefore 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?



260 atoms, is it a lot?

Feynman's question and vision

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."

"How can you simulate the quantum mechanics? ... Can you do it with a new type of computer - a quantum computer? It is not a Turing machine, but a machine of a different kind".

Feynman's question and vision

Richard Feynman (1981):

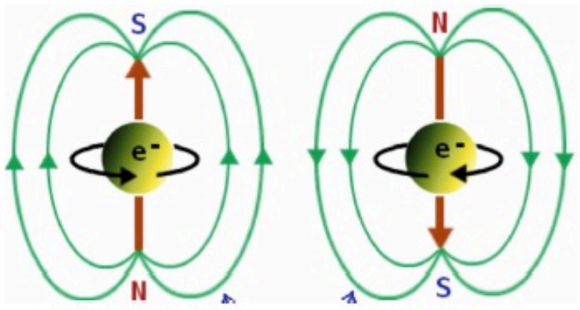
"...trying to find a computer simulation of physics, seems to me to be an excellent program to follow out...and I'm not happy with all the analyses that go with just the classical theory, because *nature isn't classical*, dammit, and if you want to make a simulation of nature, you'd better *make it quantum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

"How can you simulate the quantum mechanics? ... Can you do it with a new type of computer - a quantum computer? It is not a Turing machine, but a machine of a different kind".

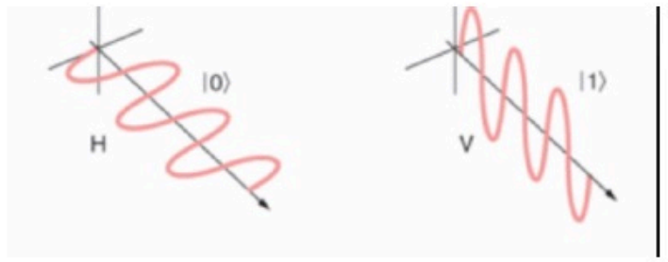
This opened the way for the idea of quantum algorithms (Deutsch '85, Deutsch-Jozsa '87, Bernstein-Vazirani '88, Shor '94)

Qubit: the two state quantum mechanical system obeying a superposition principle

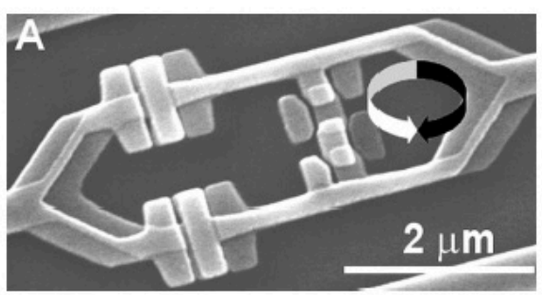
Az elektron (vagy egy atommag) mágneses dipólusmomentuma:



A fény polarizációja:



Szupravezetők fluxusa, áramiránya:



Quantum bits

1. *quantum bit, qubit, q-bit, qbit*: two-level quantum system
2. state of a qubit: $|\psi\rangle = \alpha_0 |0\rangle + \alpha_1 |1\rangle$
3. α_0, α_1 are called *amplitudes*; they are complex numbers
4. $|0\rangle$ and $|1\rangle$ are the *qubit basis states*
5. normalization condition: $|\alpha_0|^2 + |\alpha_1|^2 = 1$
6. alternative notation (*vector notation* or *spinor notation*):

$$|0\rangle \equiv \begin{pmatrix} 1 \\ 0 \end{pmatrix}, |1\rangle \equiv \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \alpha_0 |0\rangle + \alpha_1 |1\rangle \equiv \alpha_0 \begin{pmatrix} 1 \\ 0 \end{pmatrix} + \alpha_1 \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} \alpha_0 \\ \alpha_1 \end{pmatrix}$$

7. realizations: electron spin, nuclear spins (e.g., H-1, C-13), superconducting circuits, etc.

Measurement (`readout') of a qubit

1. $|\psi\rangle = \alpha_0 |0\rangle + \alpha_1 |1\rangle$
2. the probability of measuring 0 is $P_0 = |\alpha_0|^2$
3. the probability of measuring 1 is $P_1 = |\alpha_1|^2 = 1 - P_0$
4. if the outcome of the measurement is 0, then the state changes to $|0\rangle$
5. if the outcome of the measurement is 1, then the state changes to $|1\rangle$

More qubits

1. states of two qubits: $|\psi\rangle = \alpha_{00}|00\rangle + \alpha_{01}|01\rangle + \alpha_{10}|10\rangle + \alpha_{11}|11\rangle$
2. normalization condition: $\sum_{x\in\{0,1\}^2} |\alpha_x|^2 = 1$
3. a single-qubit state can be represented on the Bloch sphere; does not work for multiple-qubit states
4. measurement of one qubit: e.g., of the first one: $P_0 = |\alpha_{00}|^2 + |\alpha_{01}|^2$, and the post-measurement state after measuring 0 is

$$|\psi_{\text{pm}}\rangle = \frac{\alpha_{00}|00\rangle + \alpha_{01}|01\rangle}{\sqrt{P_0}}$$

5. example for a two-qubit product state:

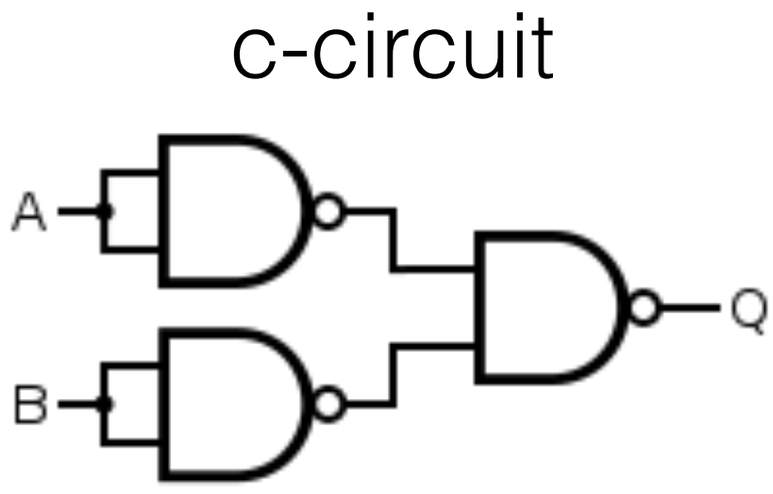
$$|\psi\rangle = \frac{1}{2}|00\rangle + \frac{1}{2}|01\rangle + \frac{1}{2}|10\rangle + \frac{1}{2}|11\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}} \otimes \frac{|0\rangle + |1\rangle}{\sqrt{2}}$$

6. example for a two-qubit entangled state:

$$|\psi\rangle = \frac{|00\rangle + |11\rangle}{\sqrt{2}}$$

7. the state of n qubits is described by 2^n amplitudes

1-qubit quantum gates



1. q-circuit: an arrangement of "wires" and quantum gates
2. q-gates: unitary operations on a few qubits (reversible, unlike c-gates)
3. 1-qubit gate example: q-NOT (usually called the X *gate*):

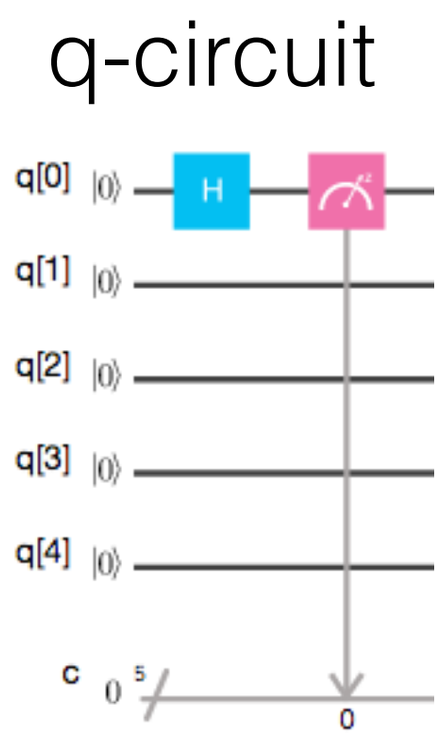
$$|\psi_1\rangle = \alpha\,|0\rangle + \beta\,|1\rangle \;\mapsto\; |\psi_2\rangle = \alpha\,|1\rangle + \beta\,|0\rangle$$

matrix representation of this gate: $X \equiv \sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$

4. further 1-qubit gate examples:

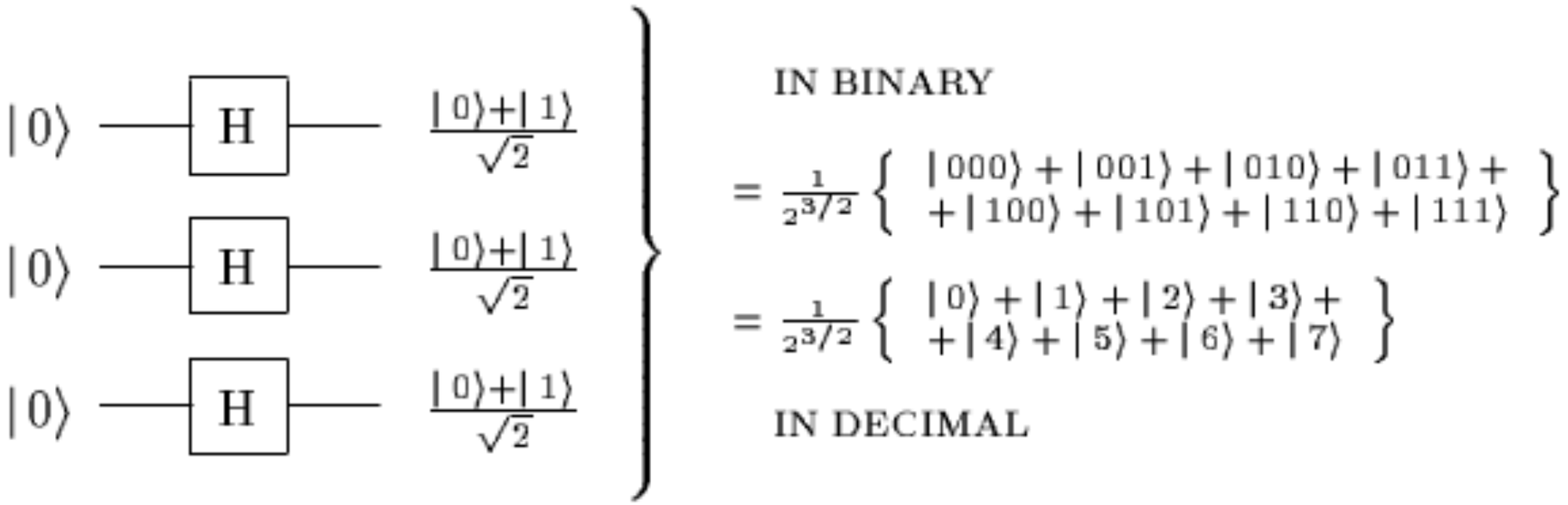
$$\text{Z gate: } Z = \sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

$$\text{Hadamard gate: } H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$



5. each 1-qubit gate generates a bijective map of the Bloch sphere to itself
6. exercise: determine the transformations generated by 1-qubit gates listed above

Creating a uniform superpositions with Hadamard Gates



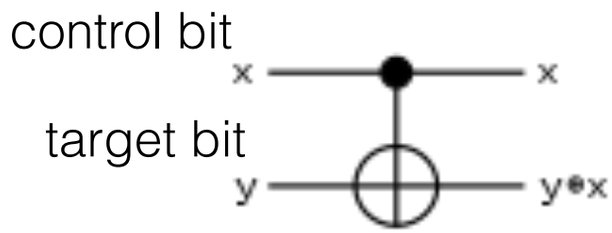
IN BINARY

$$= \frac{1}{2^{3/2}} \left\{ \begin{array}{l} |000\rangle + |001\rangle + |010\rangle + |011\rangle + \\ +|100\rangle + |101\rangle + |110\rangle + |111\rangle \end{array} \right\}$$

$$= \frac{1}{2^{3/2}} \left\{ \begin{array}{l} |0\rangle + |1\rangle + |2\rangle + |3\rangle + \\ +|4\rangle + |5\rangle + |6\rangle + |7\rangle \end{array} \right\}$$

IN DECIMAL

2-qubit quantum gates

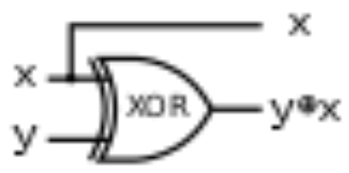


input		output	
x	y	x	y+x
|0⟩	|0⟩	|0⟩	|0⟩
|0⟩	|1⟩	|0⟩	|1⟩
|1⟩	|0⟩	|1⟩	|1⟩
|1⟩	|1⟩	|1⟩	|0⟩

1. 2-qubit gate example: *controlled-NOT* or *CNOT*
   with the basis-state ordering $|00\rangle$, $|01\rangle$, $|10\rangle$, $|11\rangle$, it is represented by

$$U_{\mathrm{CNOT}} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

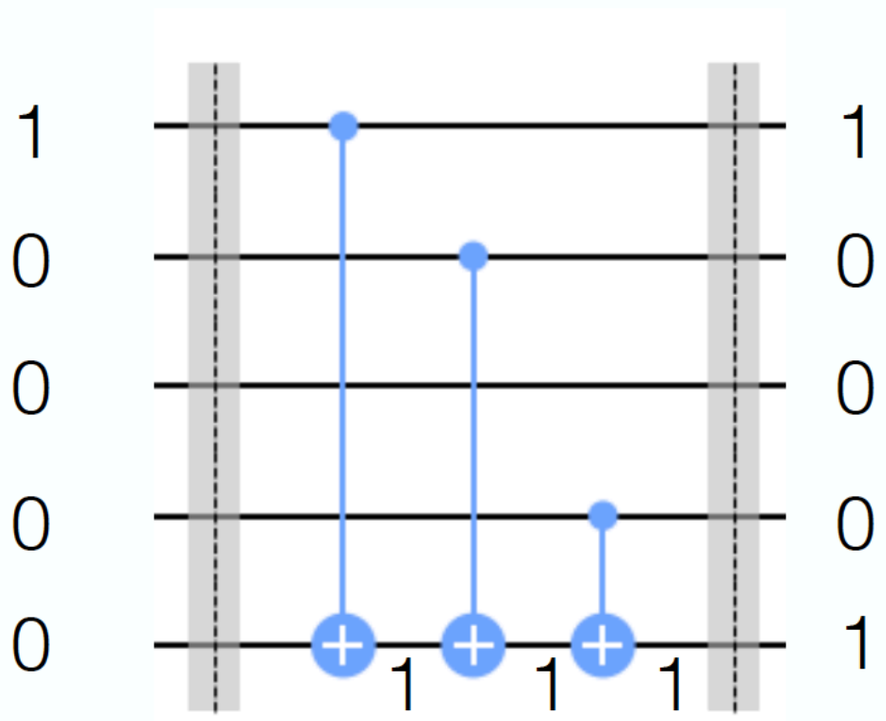
   it could be represented by a 'classical' truth table



input		output	
x	y	x	y+x
0	0	0	0
0	1	0	1
1	0	1	1
1	1	1	0

2. 1-qubit gates together with CNOT form a unversal q-gate set

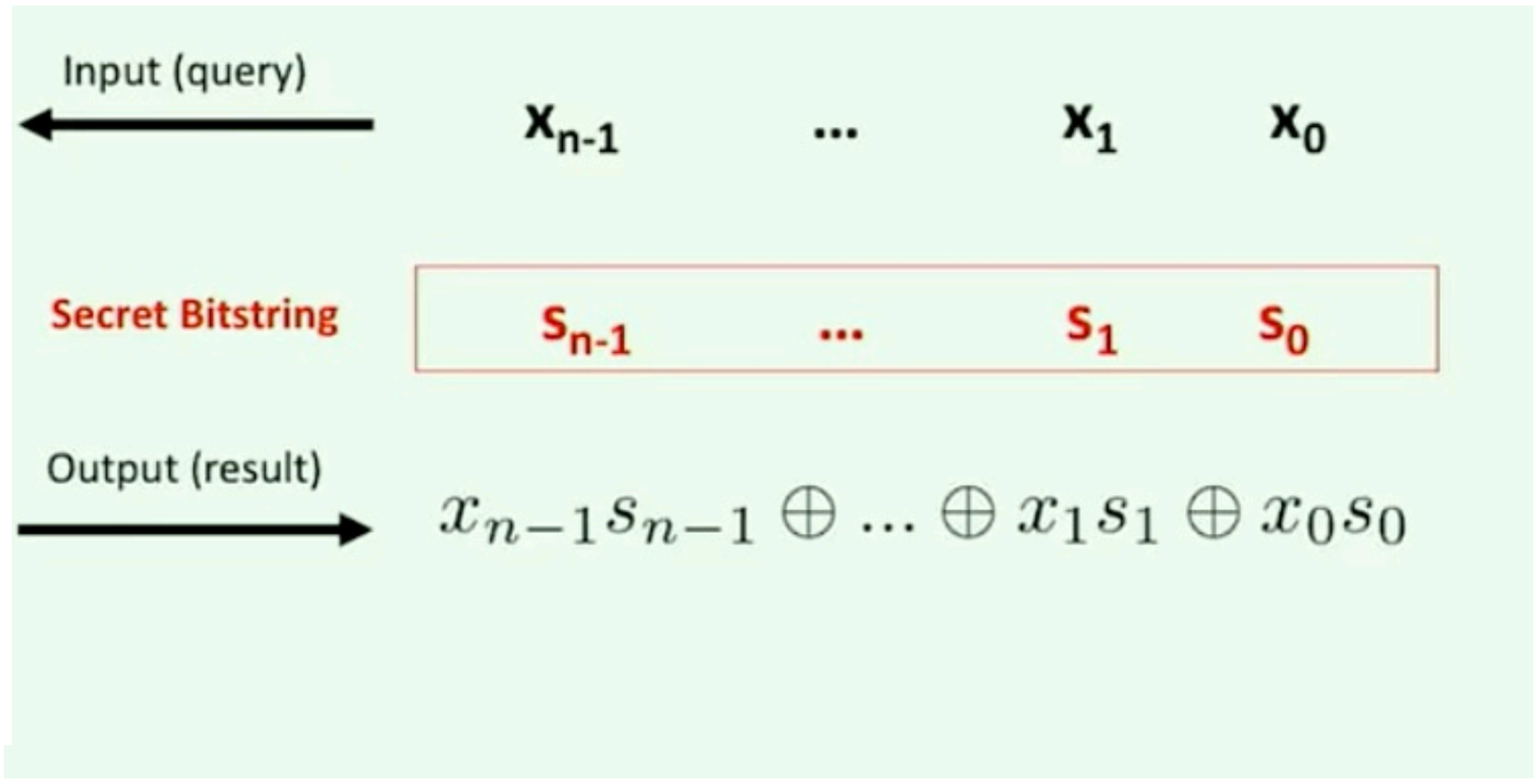
CNOTs with the same target qubit



CNOTs perform the scalar product:
if an secret bit 1 has an input 1, then the auxiliary bit is flipped

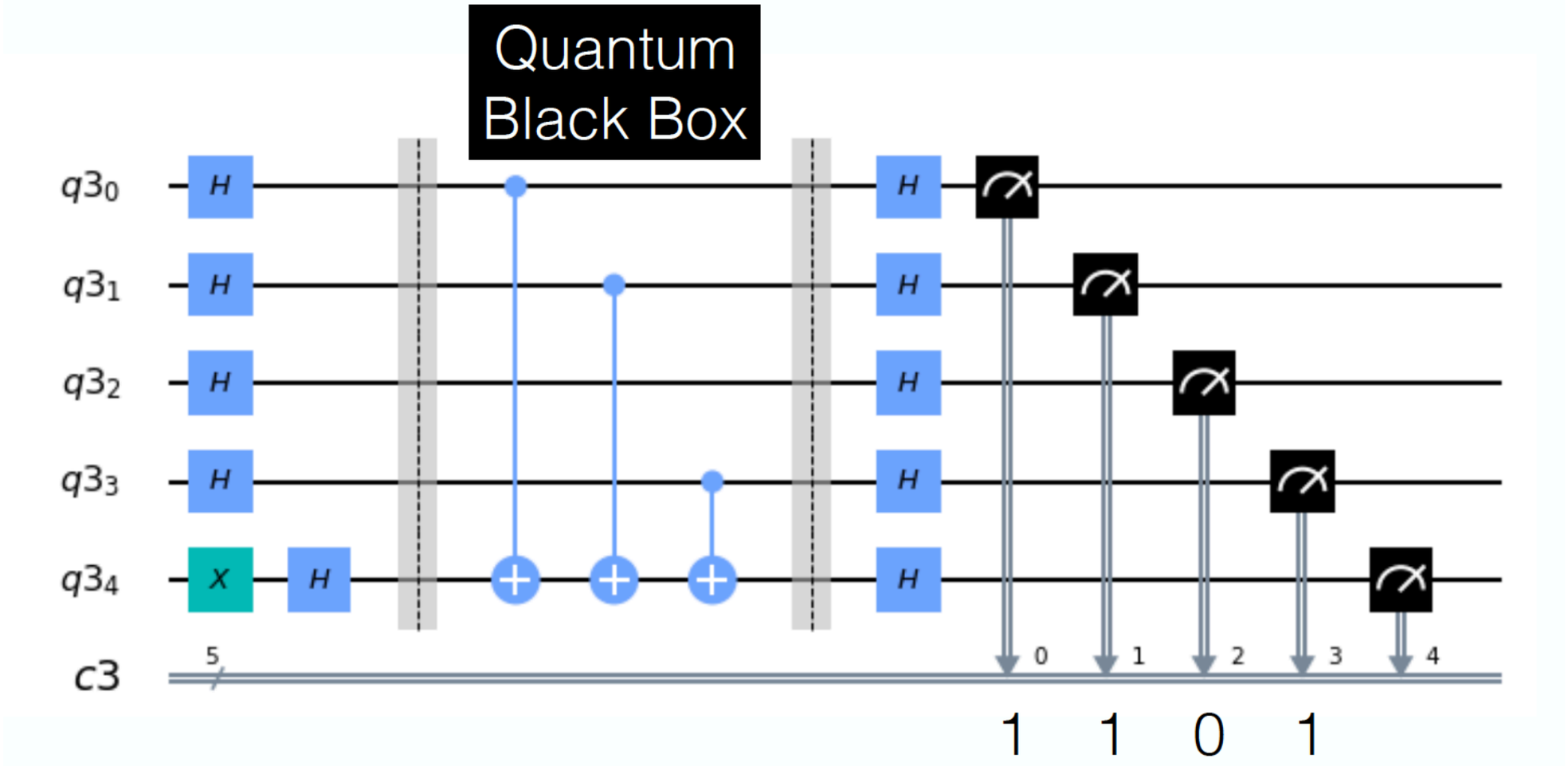
The Bernstein-Vazirani Problem

Black Box



How many queries do we need to determine the secret bit string?

The Bernstein-Vazirani Algorithm



See lecture of András

Shor's algorithm in short

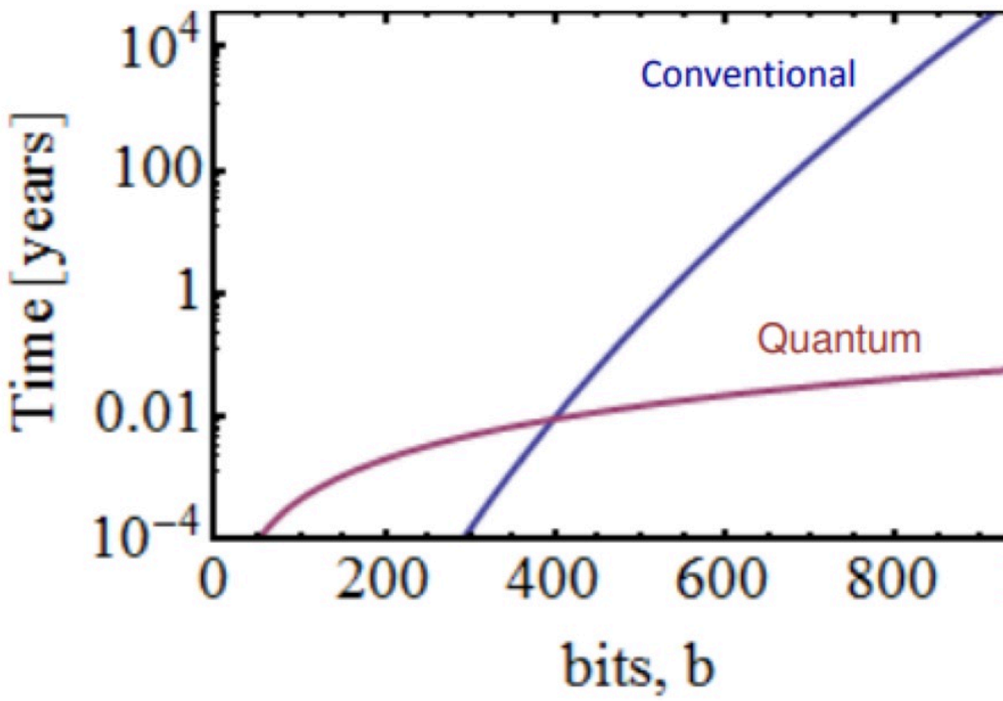
Peter Shor

Shor's algorithm is a quantum algorithm for factoring a number N in $O(n^3)$ time, named after Peter Shor.

Factor a number into primes:

M = p * q

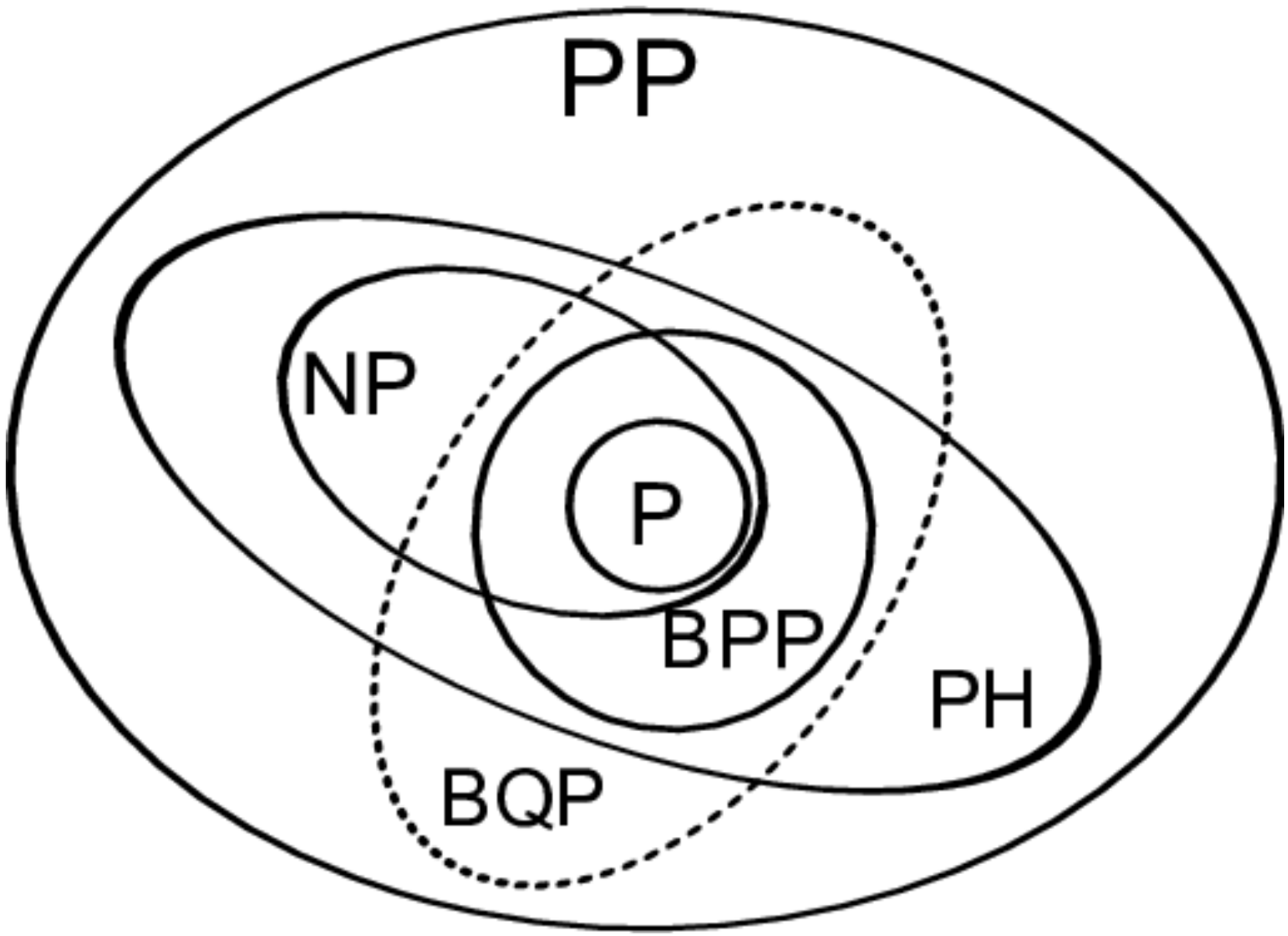
How long will it take ? (t)



Source: http://www.ibm.com

Classical	Quantum
$t \sim O(2^{n\wedge(1/2)})$	**$t \sim O(n^3)$**

Complexity classes



STRUCTURE OF THIS WORKSHOP

09:00 - 09:15 Gergely Gábor Barnaföldi (Wigner FK) -- Opening Talk

09:15 - 09:30 Péter Domokos (Wigner FK) -- Hungarian Quantum technology

09:30 - 10:45 Zoltán Zimborás (Wigner FK) -- Introduction to Quantum Computing

10:45 - 11:15 Coffee break

11:15 - 12:00 Katalin Friedl (BME) -- Quantum searching in an unsorted database: Grover's algorithm

12:00 - 12:15 Zsolt Tabi (ELTE) -- Quadratic unconstrained binary optimization (QUBO) problems in classical computer science

12:15 - 12:45 Gábor Vattay (ELTE) -- QUBO problems and adiabatic quantum computing

13:45 - 14:15 Péter Vrana (BME) -- Anyonic Quantum Computing

14:15 - 15:00 András Pályi (BME) -- Introduction to Qiskit

15:00 - 15:30 Coffee break

15:30 - 17:00 Ákos Budai, András Pályi, Zoltán Zimborás -- Hands-on Session: Running experiments on IBM's Quantum Computer

November 22. Friday, lectures:

09:00 - 09:45 Mátyás Koniorczyk -- Adiabatic Quantum Computing and QUBO from the point of view of practical Operations Research

09:45 - 10:30 Mátyás Koniorczyk -- Hands-on Session: D-Wave's Ocean Software

10:30 - 11:00 Coffee break

11:00 - 13:00 Zoltan Zimboras, Ákos Budai -- Advanced practice with Qiskit: Grover's search and Shor's Algorithm