Quantum computing in practice

& applications to cryptography

Renaud Lifchitz

OPPIDA

NoSuchCon, November 19-21, 2014

Speaker's bio

- French senior security engineer working at Oppida (`http://www.oppida.fr`), France
- Main activities:
  - Penetration testing & security audits
  - Security research
  - Security trainings
- Main interests:
  - Security of protocols (authentication, cryptography, information leakage, zero-knowledge proofs...)
  - Number theory (integer factorization, primality testing, elliptic curves...)

Goals of this talk

- Introduce quantum physics basics to newcomers
- Give “state-of-the-art” results in quantum computing & cryptography
- Explain principles and basic blocks to build quantum circuits
- Give people ideas, tools and hardware access to practice quantum computing

Outline

Section 1

Basics of quantum computing

Quantum principles

1. Small-scale physical objects (atom, molecule, photon, electron, ...) both behave as particles and as waves during experiments (quantum duality principle)
2. Main characteristics of these objects (position, spin, polarization, ...) are not determined, have multiple values according to a probabilistic distribution (quantum superposition principle / Heisenberg's uncertainty principle)
3. Further interaction or measurement will collapse this probability distribution into a single, steady state (quantum decoherence principle)
4. Consequently, copying a quantum state is not possible (no-cloning theorem)

- We can take advantage of the first 3 principles to do powerful non-classical computations

Quantum principles

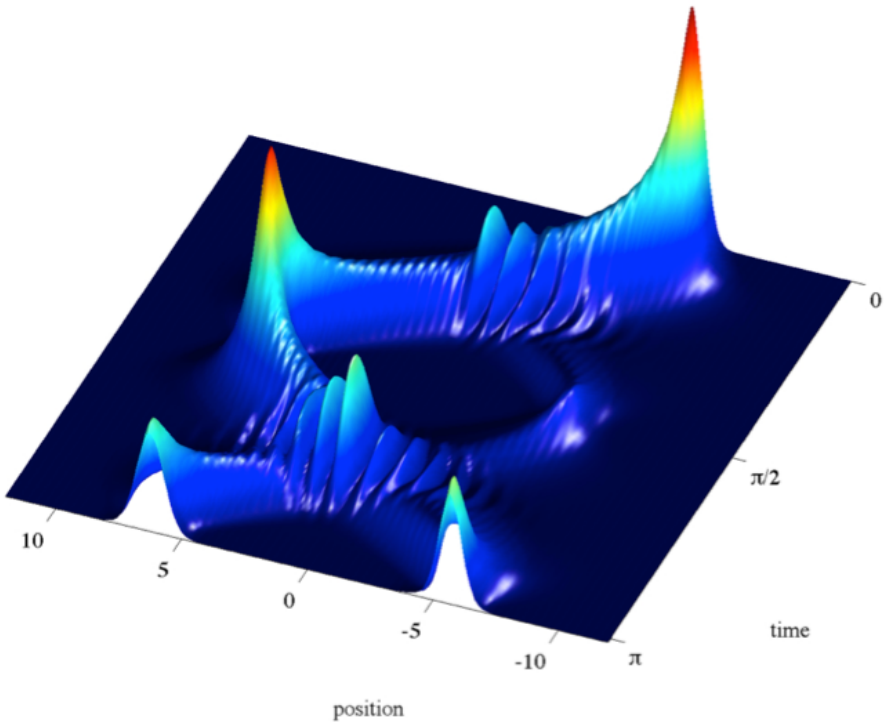


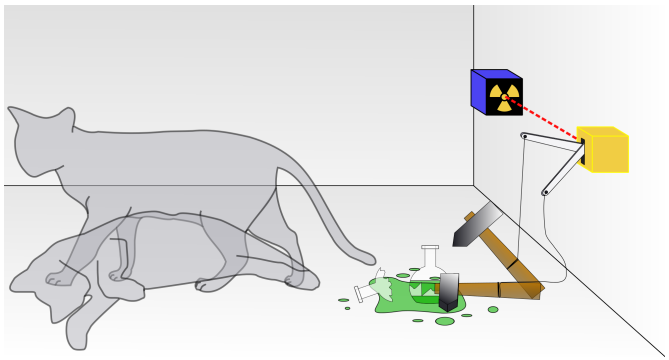
Figure : Position of an atom under quantum conditions across time, sometimes it is 100% determined, sometimes 50% - *Image created by Thomas Fogarty, graduate student from University College Cork in Ireland*

Recent quantum experiments

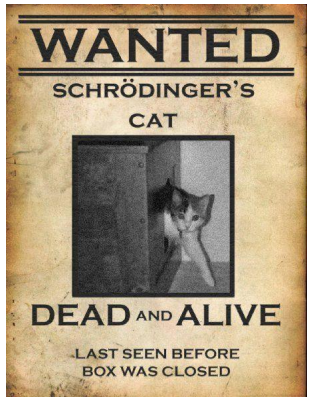
- Instant interaction of entangled qubits - EPR Paradox: Summer 2008, University of Geneva, Nicolas Gisin and his colleagues determined that the speed of the quantum interaction is at least 10000 times the speed of light using correlated photons at a 18-km distance (`http://arxiv.org/abs/0808.3316`)

- Quantum teleportation: September 2014, same team of scientists successfully achieved a 25-km quantum teleportation

Schrödinger's cat though experiment



- Paradox, though experiment, designed by Austrian physicist Erwin Schrödinger in 1935
- A cat, a bottle of poison, a radioactive source, and a radioactivity detector are placed in a sealed box
- If the detector detects radioactivity, the bottle is broken, killing the cat
- Until we open the box, the cat may be both alive **AND** dead!



Current freely available quantum systems

- Quantum number generator:
  - Commercial ID Quantique “Quantis” provides 4 Mbits/s to 16 MBits/s of true quantum randomness:

USB


PCI Express


PCI

  - Online “Quantum Random Bit Generator” (QRBG121) service: `http://random.irb.hr/`
- Quantum encryption system:
  - Commercial ID Quantique “Cerberis” & “Centauris” allow Quantum Key Distribution (QKD) and encryption up to 100 Gbps and 100 km:

Quantum cryptography

- Unbreakable cryptography, even with a quantum computer
- Doesn't rely on math problems we don't know how to solve, but on laws of physics we can't get around
- Uses QKD (Quantum Key Distribution) to exchange a symmetric key of the same size as the message (one-time pad) over a quantum channel (optical fiber for instance)
- The encrypted message is sent classically
- Interception is useless: the attacker will alter half of the key bits on average, and the receiver will detect the snooping thanks to quantum error correction codes

- Quantum Key Distribution networks exist in Geneva (Switzerland), Vienna (Austria), Massachusetts (USA), Tokyo (Japan) for banking or academic purposes
- Max distance is about 100 kms

Current coherence times of qubits

Qubit type	Coherence time
Silicon nuclear spin	25 s.
Trapped ion	15 s.
Trapped neutral atom	10 s.
Phosphorus in silicon	10 s.
NMR molecule nuclear spin	2 s.
Photon (infrared photon in optical fibre)	0.1 ms.
Superconducting qubit	4 μs.
Quantum dot	3 μs.

Figure : Current sorted coherence times of qubits (source: Institute Of Physics Publishing 2011, U.K.)

Section 2

Quantum gates and circuits

Qubit representations

- Constant qubits 0 and 1 are represented as $|0\rangle$ and $|1\rangle$
- They form a 2-dimension basis, e.g. $|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ and $|1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$
- An arbitrary qubit q is a linear superposition of the basis states: $|q\rangle = \alpha|0\rangle + \beta|1\rangle = \begin{bmatrix} \alpha \\ \beta \end{bmatrix}$ where $\alpha \in \mathbb{C}$, $\beta \in \mathbb{C}$
- When q is measured, the real probability that its state is measured as $|0\rangle$ is $|\alpha|^2$ so $|\alpha|^2 + |\beta|^2 = 1$
- Combination of qubits forms a quantum register and can be done using the tensor product: $|10\rangle = |1\rangle \otimes |0\rangle = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$
- First qubit of a combination is usually the least significant qubit of the quantum register
- A qubit can also be viewed as a unit vector within a sphere (Bloch sphere)

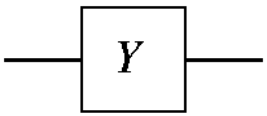
Basics of quantum gates

- For thermodynamic reasons, a quantum gate must be reversible
- It follows that quantum gates have the same number of inputs and outputs
- A n-qubit quantum gate can be represented by a 2^nx2^n unitary matrix
- Applying a quantum gate to a qubit can be computed by multiplying the qubit vector by the operator matrix on the left
- Combination of quantum gates can be computed using the matrix product of their operator matrix
- In theory, quantum gates don't use any energy nor give off any heat

Pauli-X gate

Pauli-X gate	**Number of qubits**: 1	**Symbol**: 
Description: Quantum equivalent of a NOT gate. Rotates qubit around the X-axis by Π radians. $X.X = I$.		
Operator matrix: $X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$		

Pauli-Y gate

Pauli-Y gate	**Number of qubits**: 1	**Symbol**: 
Description: Rotates qubit around the Y-axis by Π radians. $Y.Y = I$.		
Operator matrix: $Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$		

Pauli-Z gate

Pauli-Z gate	**Number of qubits**: 1	**Symbol**: 
Description: Rotates qubit around the Z-axis by Π radians. $Z.Z = I$.		
Operator matrix: $Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$		

Hadamard gate

Hadamard gate	**Number of qubits**: 1	**Symbol**: 
Description: Mixes qubit into an equal superposition of $\left\|0\right\rangle$ and $\left\|1\right\rangle$.		
Operator matrix: $H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$		

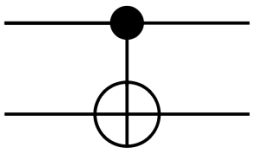
Hadamard gate

- The Hadamard gate is a special transform mapping the qubit-basis states $|0\rangle$ and $|1\rangle$ to two superposition states with "50/50" weight of the computational basis states $|0\rangle$ and $|1\rangle$:

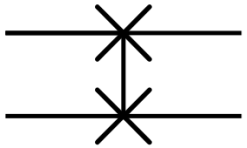
$$H.|0\rangle = \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$$
$$H.|1\rangle = \frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle$$

- For this reason, it is widely used for the first step of a quantum algorithm to work on all possible input values in parallel

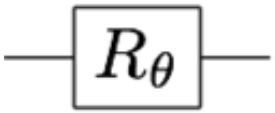
CNOT gate

CNOT gate	**Number of qubits**: 2	**Symbol**: 
Description: Controlled NOT gate. First qubit is control qubit, second is target qubit. Leaves control qubit unchanged and flips target qubit if control qubit is true.		
Operator matrix: $CNOT = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$		

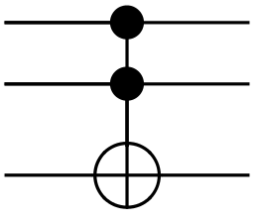
SWAP gate

SWAP gate	**Number of qubits**: 2	**Symbol**: 
Description: Swaps the 2 input qubits.		
Operator matrix: $SWAP = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$		

Phase shift gate

Phase shift gate	**Number of qubits**: 1	**Symbol**: 
Description: Family of gates that leave the basis state $\|0\rangle$ unchanged and map $\|1\rangle$ to $e^{i\theta}\|1\rangle$.		
Operator matrix: $R_\theta = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\theta} \end{bmatrix}$		

Toffoli gate

Toffoli gate	**Number of qubits**: 3	**Symbol**: 

Description: Controlled-Controlled-NOT gate. First 2 qubits are control qubits, third one is target qubit. Leaves control qubits unchanged and flips target qubit if both control qubits are true.

Operator matrix: $CCNOT = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$

Universal gates

A set of quantum gates is called **universal** if any classical logic operation can be made with only this set of gates. Examples of universal sets of gates:

- Hadamard gate, Phase shift gate (with $\theta = \frac{\Pi}{4}$ and $\theta = \frac{\Pi}{2}$) and Controlled NOT gate
- Toffoli gate only

Circuit designing challenges

- Qubits and qubit registers cannot be copied in any way
- In simulation like in reality, number of used qubits must be limited (qubit reuse wherever possible)
- Qubit registers shifts are costly, moving gates “reading heads” is somehow easier
- In reality, quantum error codes should be used to avoid partial decoherence during computation

Section 3

Fundamental quantum algorithms

Grover's algorithm

- Pure quantum algorithm for searching among N unsorted values
- Complexity: $O(\sqrt{N})$ operations and $O(\log N)$ storage place
- Probabilistic, iterating and optimal algorithm

Quantum Fourier Transform (QFT) algorithm

- Quantum equivalent to the classical discrete Fourier Transform algorithm
- Finds periods in the input superposition
- Only requires $O(n^2)$ Hadamard gates and controlled phase shift gates, where n is the number of qubits

Shor's algorithm

- Pure quantum algorithm for integer factorization that runs in polynomial time formulated in 1994
- Complexity: $O((\log N)^3)$ operations and storage place
- Probabilistic algorithm that basically finds the period of the sequence $a^k \mod N$ and non-trivial square roots of unity $\mod N$
- Uses QFT
- Some steps are performed on a classical computer

Section 4

Attacks against cryptography

Breaking asymmetric cryptography

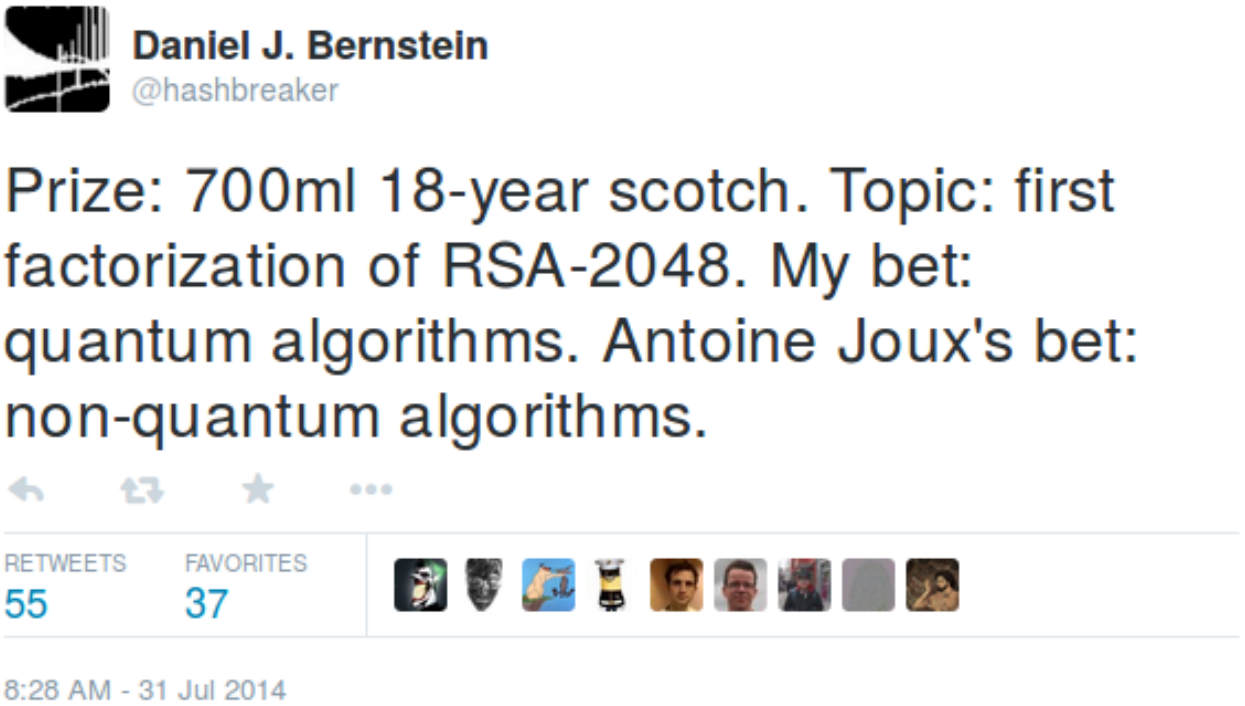
- Most asymmetric cryptosystems rely on the integer factorization difficulty
- Shor's algorithm is able to factor integers efficiently and similar algorithms exist for solving discrete logarithms
- RSA and Diffie–Hellman key exchange are quite easily broken
- HTTPS, SSL, SSH, VPNs and certificates security will be seriously threatened

Current records are RSA factorization of 21 in October 2012 (real quantum computation), and factorization of 143 in April 2012 (adiabatic quantum computation).

Breaking symmetric cryptography

- It is possible to test multiple symmetric keys in parallel with a quantum algorithm
- More precisely, using Grover's algorithm, we can test N keys in $\sqrt{N}$ steps
- This divides at least all current keylength strengths by 2

The new RSA-2048 challenge



Section 5

Quantum computing simulations & tools

Quantum Circuit Simulator (Android)

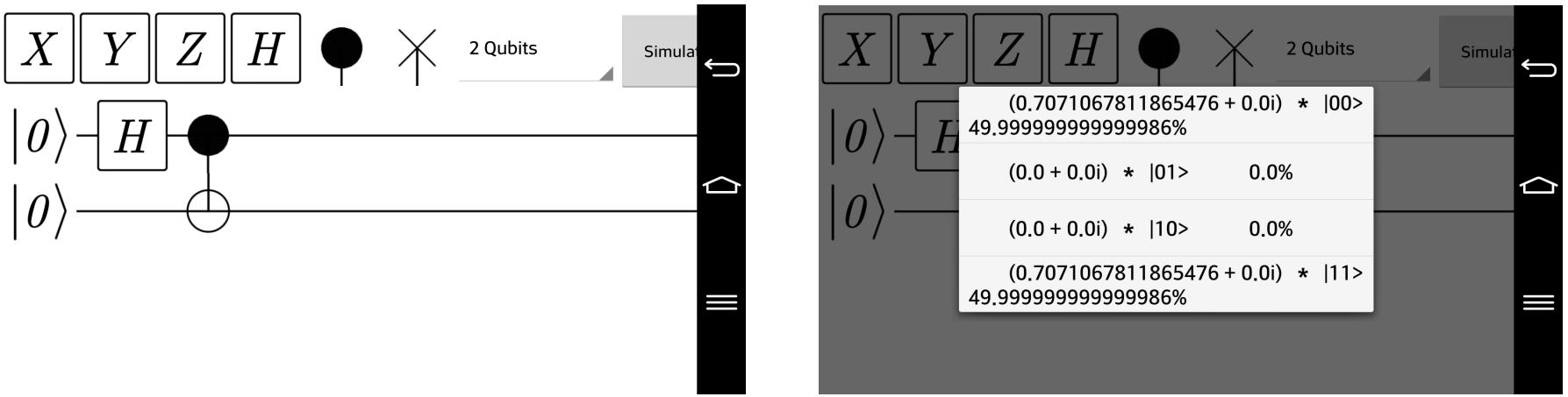


Figure : Design and simulation of a qubit entanglement circuit. Those 2 qubits can interact instantly at any distance according to the nonlocality principle.

QCL

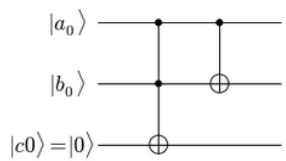
```
if number mod 2 == 0 { exit "number must be odd"; }
if testprime(number) { exit "prime number"; }
if testprimepower(number) { exit "prime power"; };
{
  {                              // generate random base
    x=floor(random()*(number-3))+2;
  } until gcd(x,number)==1;
  print "chosen random x =",x;
  H(reg1);                       // Hadamard transform
  expn(x,number,reg1,reg2);      // modular exponentiation
  measure reg2;                  // measure 2nd register
  dft(reg1);                     // Fourier transform
  measure reg1,m;                // measure 2st register
  reset;                         // clear local registers
  if m==0 {                      // failed if measured 0
    print "measured zero in 1st register. trying again ...";
  } else {
    c=m*0.5^(2*width);           // fixed point form of m
    q=denominator(c,qmax);       // find rational approximation
    p=floor(q*c+0.5);
    print "measured",m,", approximation for",c,"is",p,"/",q;
    if q mod 2==1 and 2*q<qmax { // odd q ? try expanding p/q
      print "odd denominator, expanding by 2";
      p=2*p; q=2*q;
    }
    if q mod 2==1 {              // failed if odd q
      print "odd period. trying again ...";
    } else {
      print "possible period is",q;
      e=powmod(x,q/2,number);    // calculate candidates for
      a=(e+1) mod number;        // possible common factors
      b=(e+number-1) mod number; // with number
      print x,"^",q/2,"+ 1 mod",number,"=",a,",",
            x,"^",q/2,"- 1 mod",number,"=",b;
      factor=max(gcd(number,a),gcd(number,b));
    }
  }
} until factor>1 and factor<number;
print number,"=",factor,"*",number/factor;
```

```
renaud@primestation:~/installations/qcl/qcl-0.6.4-x86_64-linux-gnu$ ./qcl
QCL Quantum Computation Language (64 qubits, seed 1416240575)
[0/64] 1 |0>
qcl> include "shor.qcl";
qcl> shor(77);
: chosen random x = 57
: measured 11621 , approximation for 0.70929 is 61 / 86
: possible period is 86
: 57 ^ 43 + 1 mod 77 = 9 , 57 ^ 43 - 1 mod 77 = 7
: 77 = 7 * 11
[0/64] 1 |0>
qcl>
```

Figure : Shor's algorithm running in QCL (`http://tph.tuwien.ac.at/~oemer/qcl.html`)

Python & Sympy

```
In [1]: %pylab inline

        from sympy import init_session, init_printing
        init_printing(use_latex=True)

        from sympy.physics.quantum.qubit import Qubit, matrix_to_qubit, qubit_to_matrix, measure_all, measure_partial, measure_all_
        from sympy.physics.quantum.gate import *
        from sympy.physics.quantum.qapply import qapply
        from sympy.physics.quantum.qasm import Qasm
        from sympy.physics.quantum import TensorProduct

        Populating the interactive namespace from numpy and matplotlib

In [2]: def adder1q():
                q = Qasm()
                q.qubit('a_0') # input a
                q.qubit('b_0') # input b
                q.qubit('c0','0')
                q.toffoli('a_0','b_0','c0')
                q.cnot('a_0','b_0')
                return q

In [3]: q = adder1q(); q.plot()
```




```
In [4]: c = q.get_circuit(); c
Out[4]: CNOT_{2,1} C_{2,1} (X_0)

In [5]: measure_all(qapply(c*H(1)*H(0)*Qubit('000')))
Out[5]: [(|000⟩, 1/4), (|001⟩, 1/4), (|010⟩, 1/4), (|011⟩, 1/4)]
```

Figure : Simple 1-qubit adder with Sympy (http://docs.sympy.org/dev/modules/physics/quantum/)

Python & Sympy

Demo

- Hash design (CRC-8) with only CNOT gates

Python & Sympy

Demo

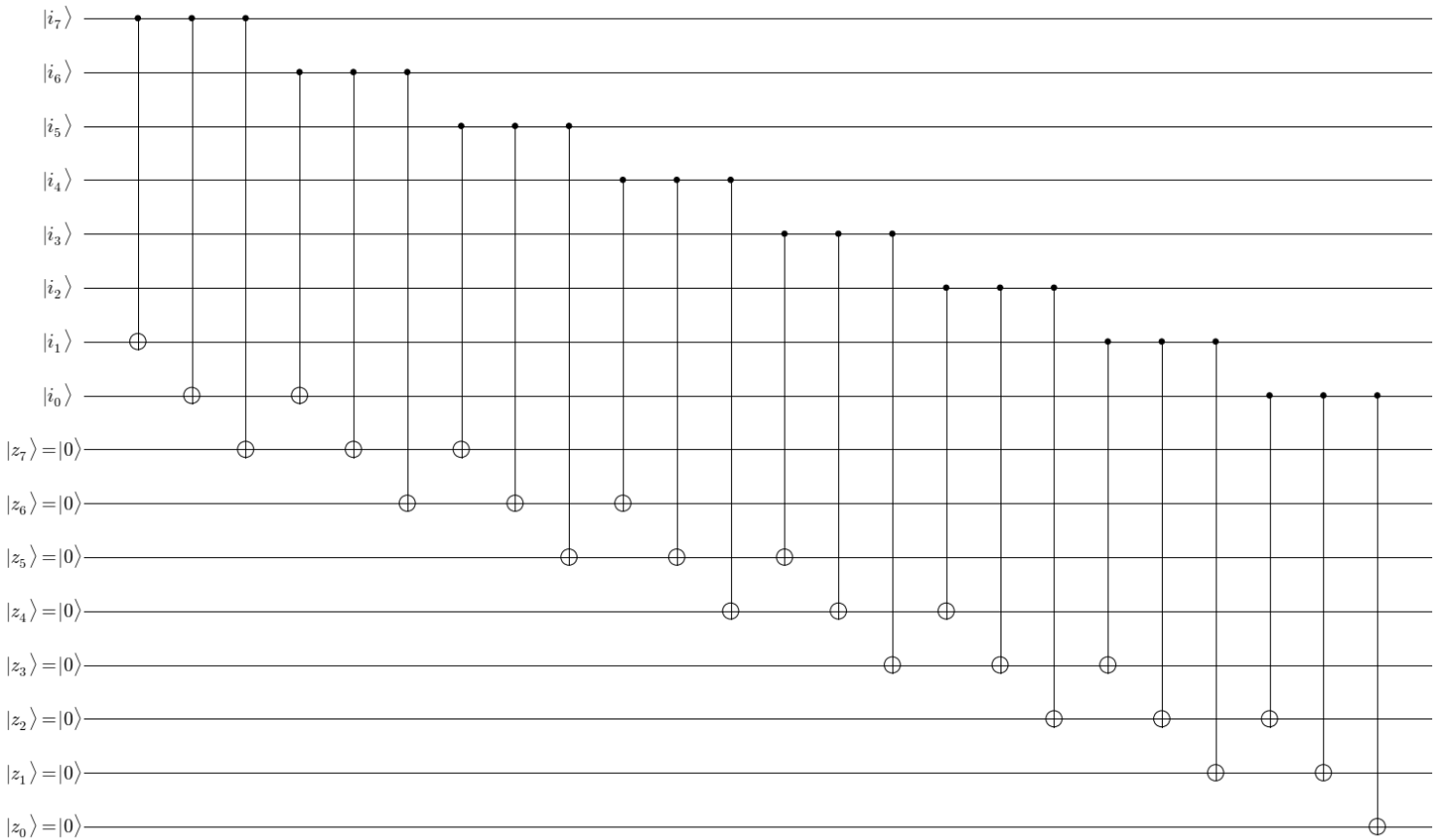


Figure : A quantum CRC-8 circuit with only CNOT gates

Quantum Computing Playground (Web)

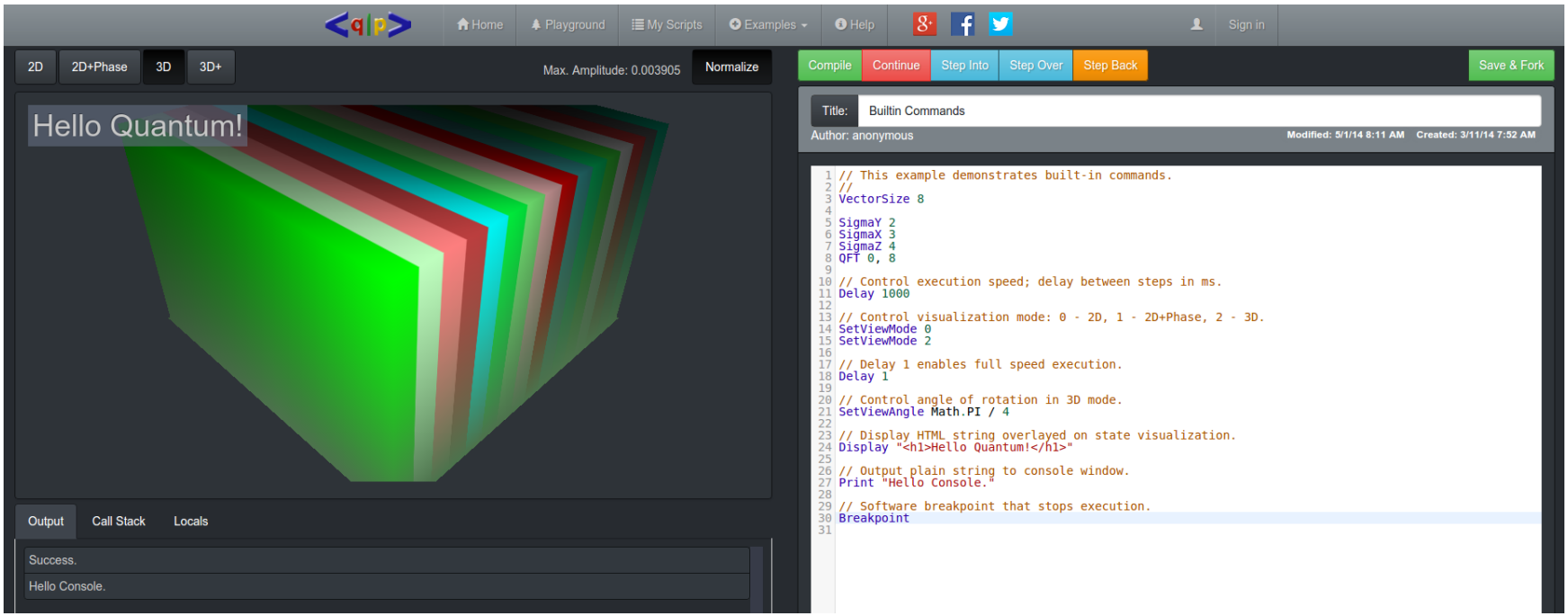


Figure : QFT on `http://www.quantumplayground.net/`

Quantum Circuit Simulator (Web) by Davy Wybiral

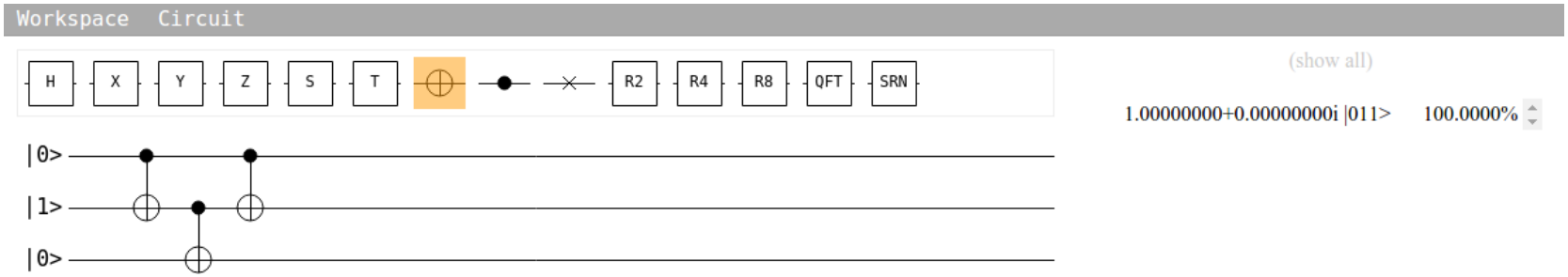


Figure : Simple 1-qubit adder on `http://www.davyw.com/quantum/`

Section 6

Computing on adiabatic quantum computers

D-Wave adiabatic computers

- D-Wave is a Canadian quantum computing company
- They have built some controversial quantum computers, D-Wave One & D-Wave Two
- D-Wave computers have been sold to Lockeed Martin and Google (shared with Nasa) for 10-15 million US dollars
- They plan to double their qubit capacity every year in the next decade

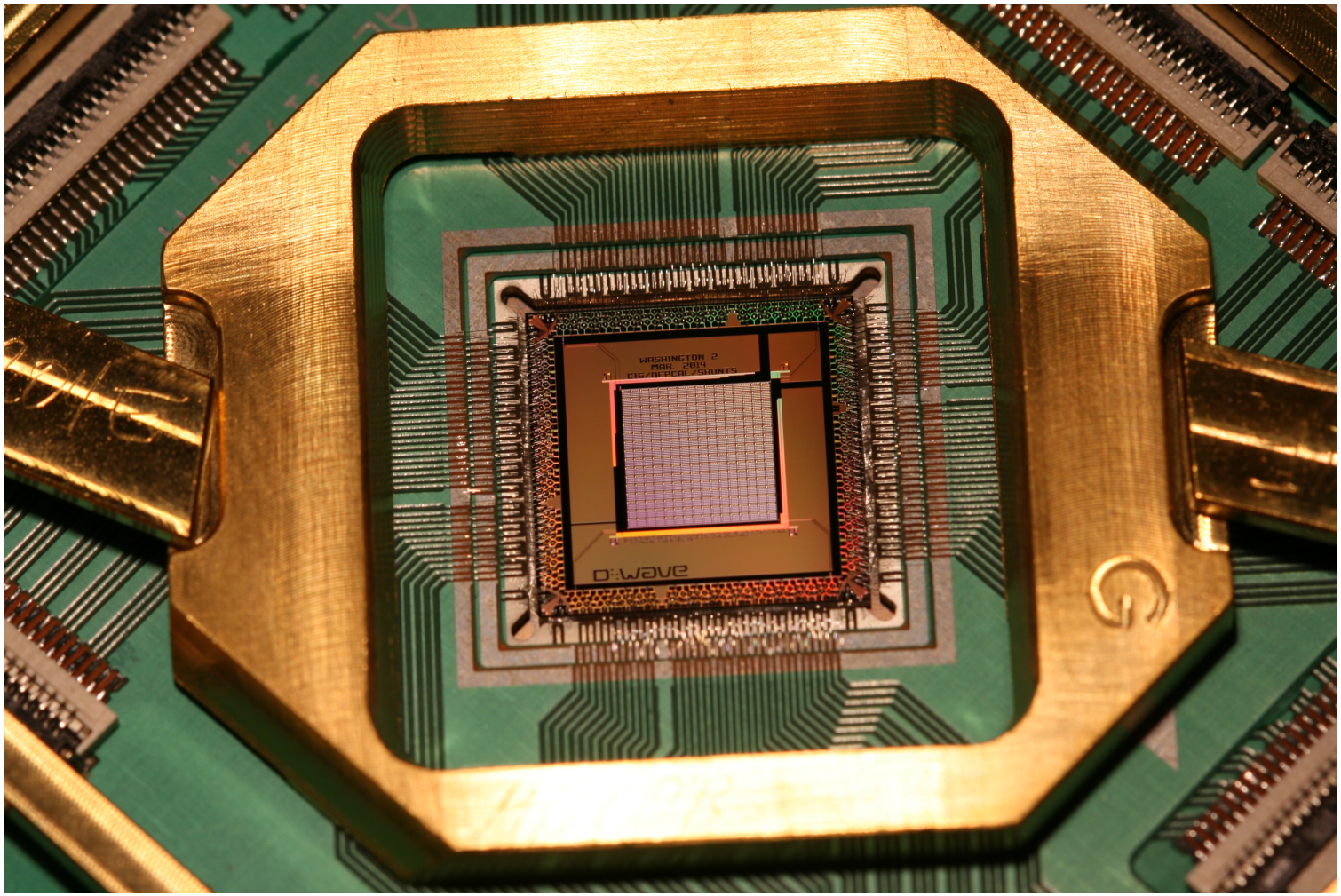


Figure : Latest D-Wave “Washington” 2048-qubit chip

How do they work?

- Probabilistic, iterating & convergent system
- A quantum state represents the solutions to the problem
- An ordinary computer will measure and rank a solution with the problem generating function G and influences the quantum state
- The quantum state will converge to a pretty good solution thanks to its thermal equilibrium and the Boltzmann probability distribution:

$$P(x_1, x_2, \ldots, x_N) = \frac{1}{Z} e^{-G(x_1, x_2, \ldots, x_N)/kT}$$

$$\text{with } Z = \sum_{k=1}^{N} \sum_{x_k = 0,1} e^{-G(x_1, x_2, \ldots, x_N)/kT}$$

I was able to factor the RSA integer 1609337 (21 bits) in 1 minute using a home-made simulation model framework (no noise).

Current limitations

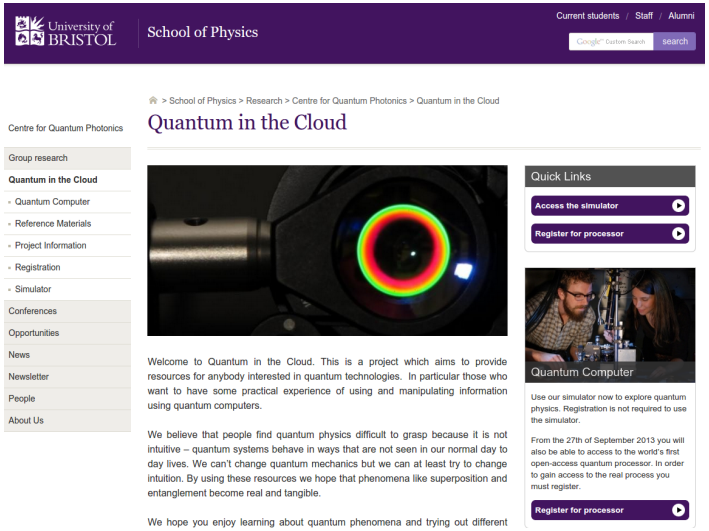
- Limited to optimization problems
- Limited to problems with solutions you can rank
- Personal opinion: better when generating function is everywhere continuous and differentiable (not the case with discrete problems like factorization)

In conclusion, adiabatic computers are specific and need to be more peer-reviewed and extensively tested to prove their real advantage.

Section 7

Computing on real quantum computers

“Quantum in the Cloud” project



- Project of the University of Bristol (U.K.), Centre for Quantum Photonics
- Full, universal, quantum computer
- Remote access (JSON/HTTP) to a 2-qubit photonic chip available upon request, 4-qubit chip available for local researchers
- Online chip simulator available for training
- Homepage: `http://www.bristol.ac.uk/physics/research/quantum/qcloud/`

“Quantum in the Cloud” project

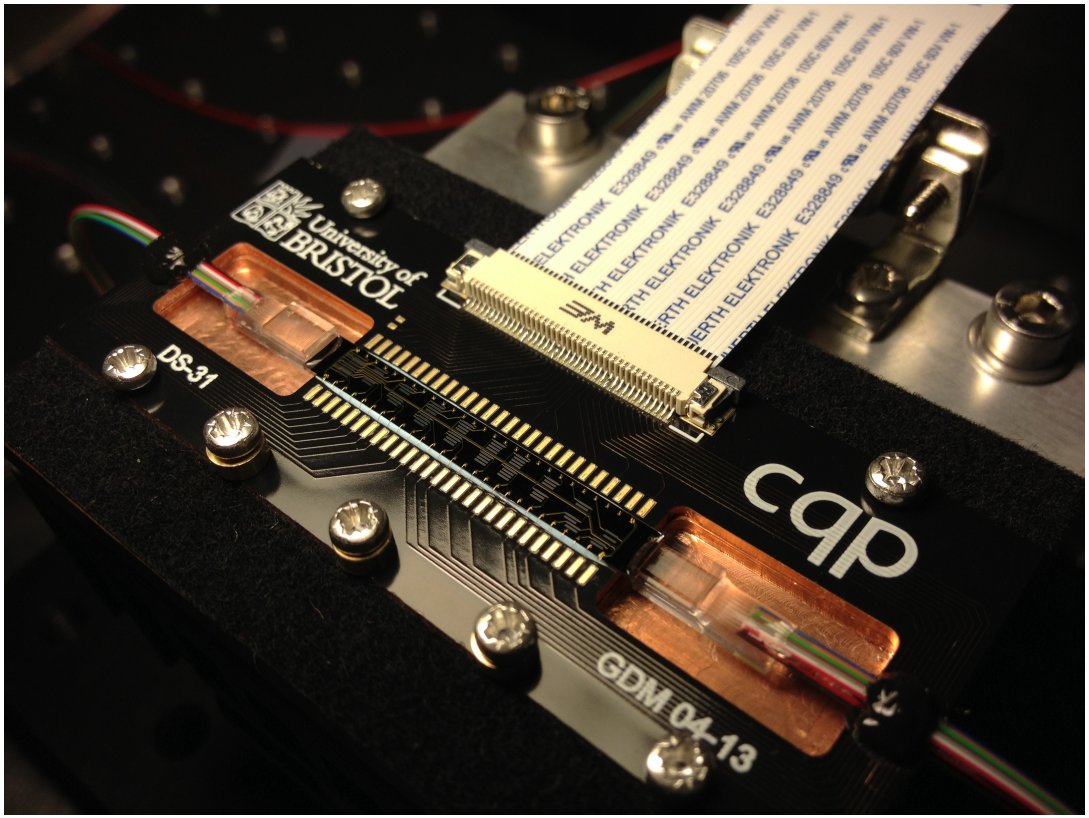
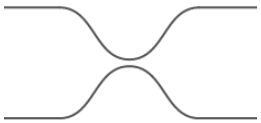


Figure : An optoelectronic quantum chip from Bristol Centre for Quantum Photonics

“Quantum in the Cloud” project

Parts of the chips

Representation	Description
	**Photon input path:** The beginning of a fiber path where you can inject photons
	**Photon output path:** The end of a fiber path where you can detect photons
	**Photon beam splitter:** A device which lets a certain fraction of light pass through it, while the rest of the light is reflected from the surface. All of the beam splitters on the CNOT-MZ chip are “50/50” beam splitters, apart from the three down the middle, which let 2/3 of the light pass through them
	**Photon phase changer:** A variable phase changer in Π radians varying from 0 to 2. In reality, a little heater that changes speed of photons.

“Quantum in the Cloud” project

Classical vs. Quantum Interference (1/3)

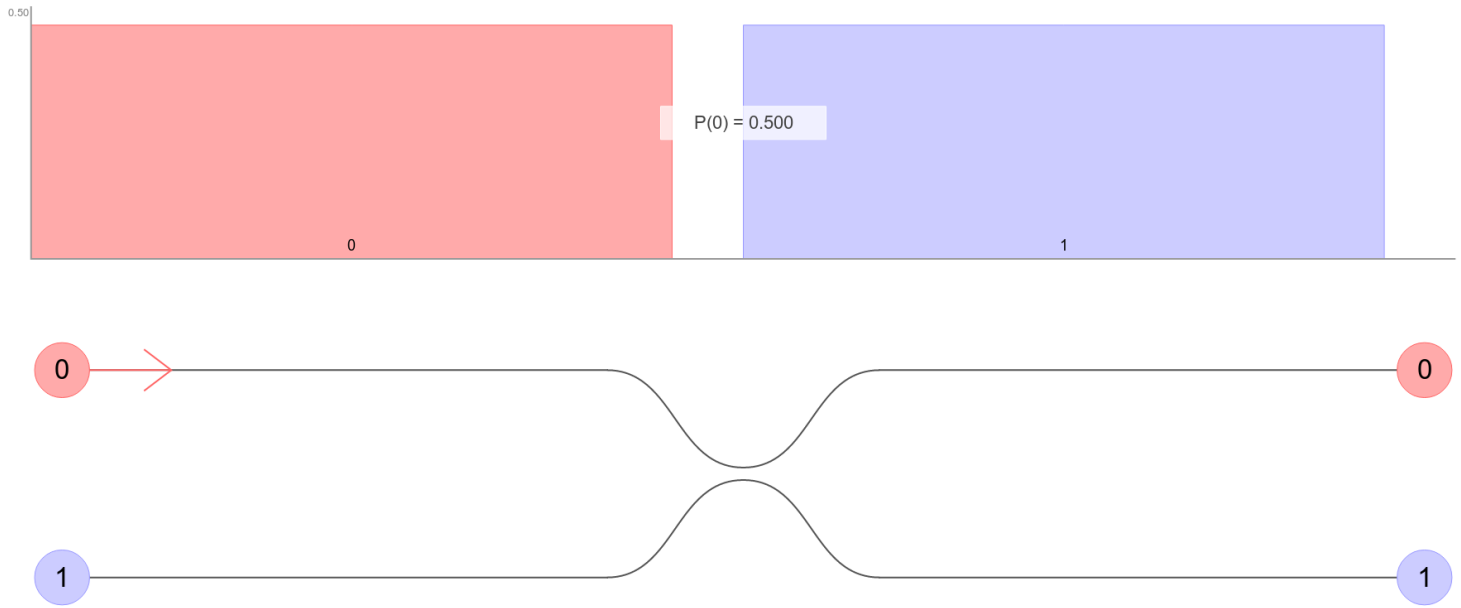


Figure : **1 input photon - classical & quantum interference**: the photon will be detected on any detector with a “50/50” probability

“Quantum in the Cloud” project

Classical vs. Quantum Interference (2/3)

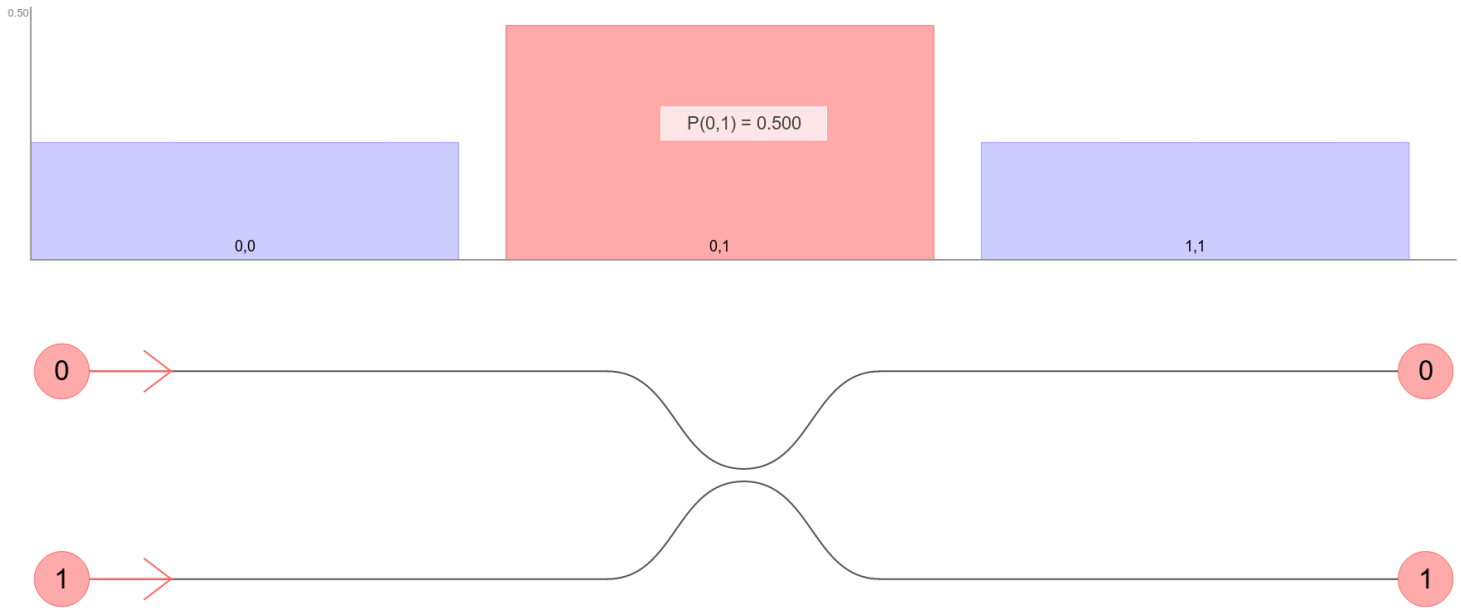


Figure : **2 input photons - classical interference**: half of the time, each detector clicks once. The other half of the time, one of the detectors clicks twice (split equally between this happening at detector 1 and detector 0)

“Quantum in the Cloud” project

Classical vs. Quantum Interference (3/3)

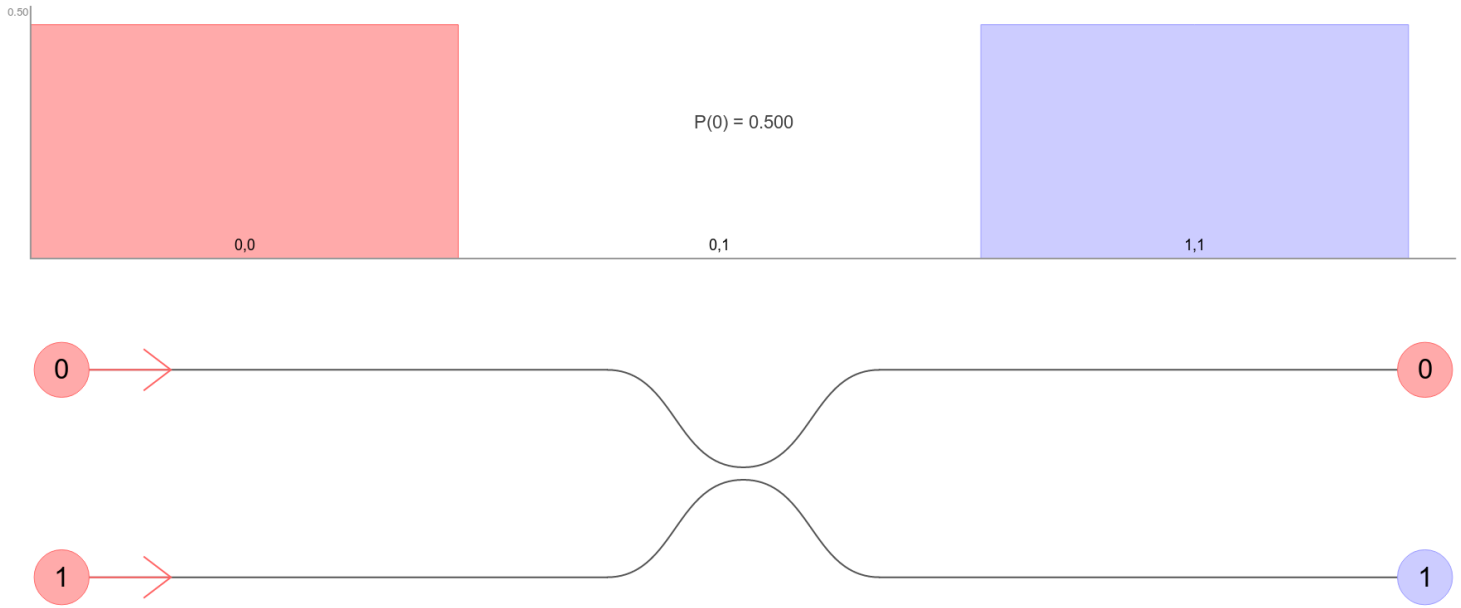
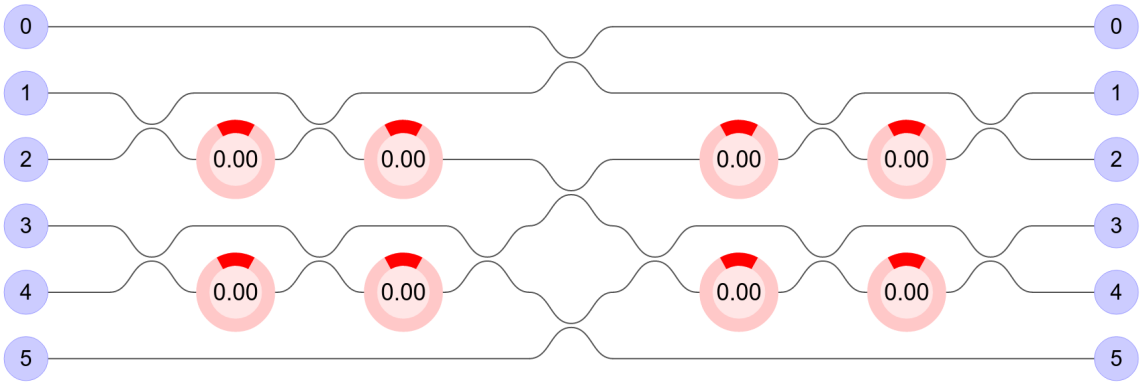


Figure : **2 input photons - quantum interference**: Both photons will “cooperate” and will always end up in the same path, causing one of the detectors to click twice. This is a purely quantum mechanical effect.

“Quantum in the Cloud” project

The CNOT-MZ chip



- 6 injection paths for a maximum of 4 photons
- 13 beam splitters
- 8 variable phase shifters

“Quantum in the Cloud” project

Postselection step

- After each experiment, some outcomes must be cancelled as their probability is not real
- Postselection is the act of restricting outcomes of a process or experiment, based on certain conditions being satisfied
- As each input qubit is coded with 2 input paths, output paths must correspond
- Outcomes with non-corresponding output paths are cancelled

“Quantum in the Cloud” project

Designing reversible logic gates with the CNOT-MZ chip

I have computed a set of possibilities for possible paths for some 1-qubit and 2-qubit gates:

	q_1	
f	0	1
id	0	1
NOT	3	4

Figure : 1-qubit gates

	q_1		q_2	
f	0	1	0	1
id	0	1	2	5
$SWAP$	0	3	2	4
$CNOT$	1	2	3	4
CMP	0	1	3	4

Figure : 2-qubit gates

“Quantum in the Cloud” project

Demo on real hardware

- NOT gate
- SWAP gate
- Quantum adder with a mixed qubit

“Quantum in the Cloud” project

Demo on real hardware - NOT gate

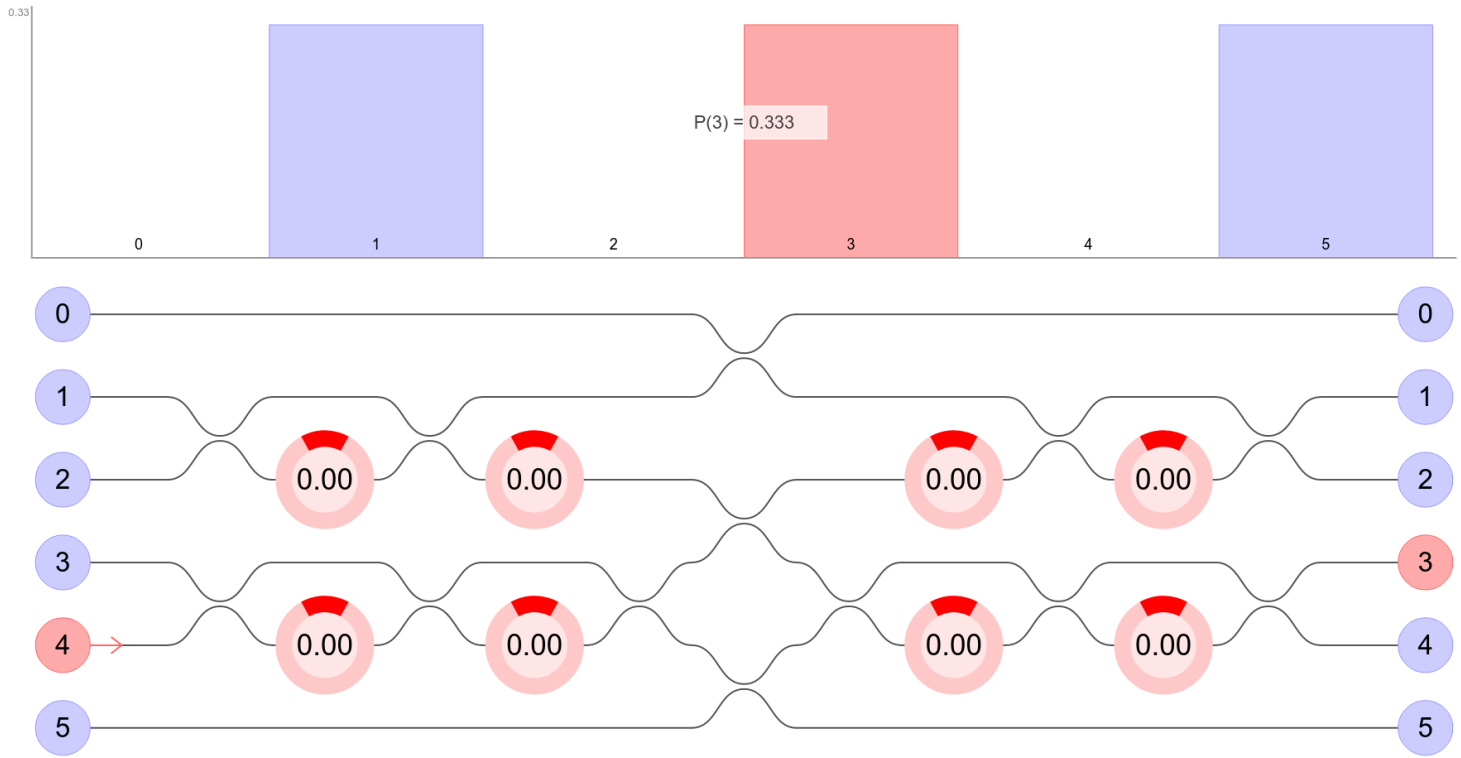


Figure : A 1-qubit NOT gate can be designed using the qubit mapping $|0\rangle \rightarrow$ (3) and $|1\rangle \rightarrow$ (4). After postselection, outcomes (1) and (5) are cancelled and we can measure that $NOT(|1\rangle) = |0\rangle$ at any time.

“Quantum in the Cloud” project

Demo on real hardware - SWAP gate

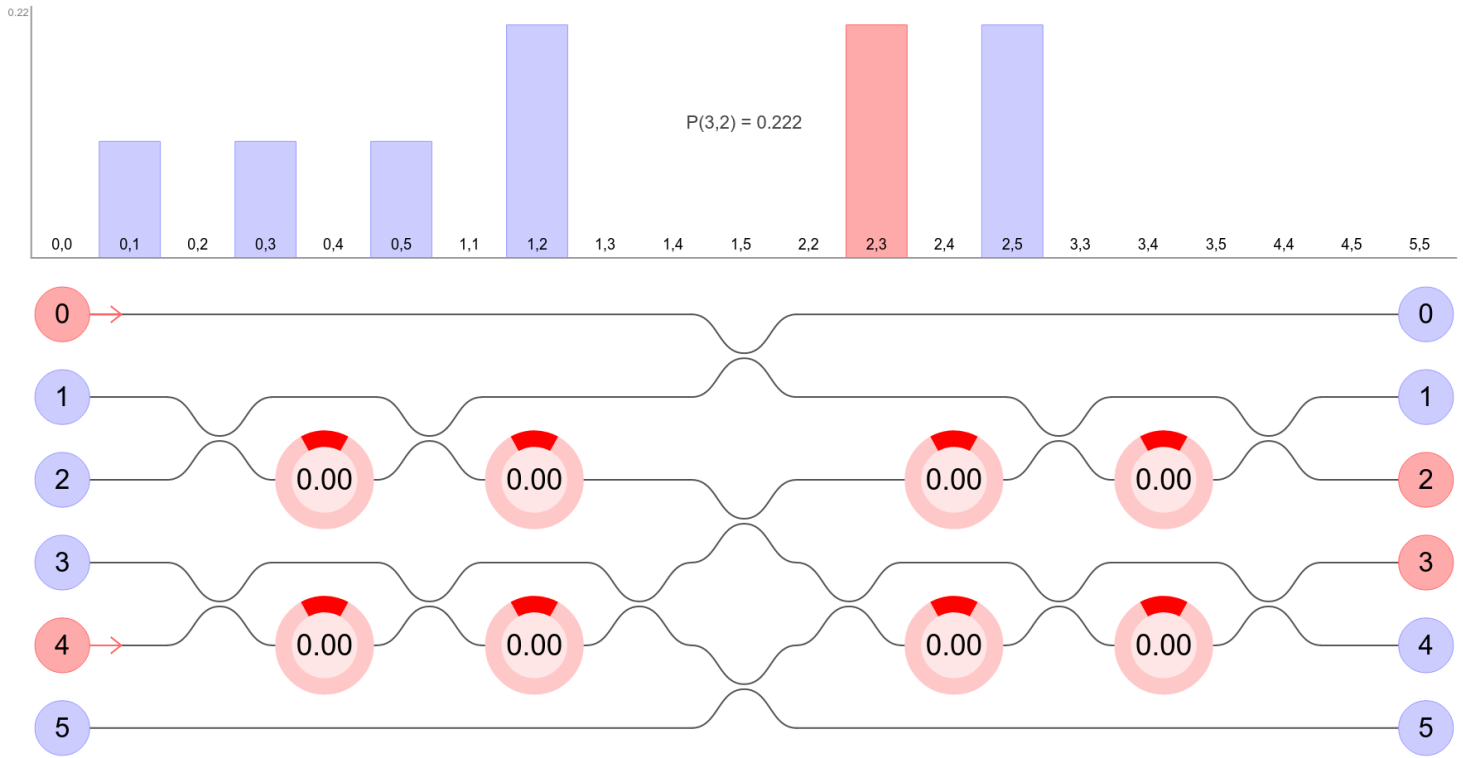


Figure : A 2-qubit SWAP gate can be designed using the qubit mapping $|0\rangle \rightarrow \textcircled{0}$ and $|1\rangle \rightarrow \textcircled{3}$ for the first qubit and $|0\rangle \rightarrow \textcircled{2}$ and $|1\rangle \rightarrow \textcircled{4}$ for the second. After postselection, we can measure that $SWAP(|01\rangle) = |10\rangle$.

“Quantum in the Cloud” project

Demo on real hardware - Quantum adder with a mixed qubit

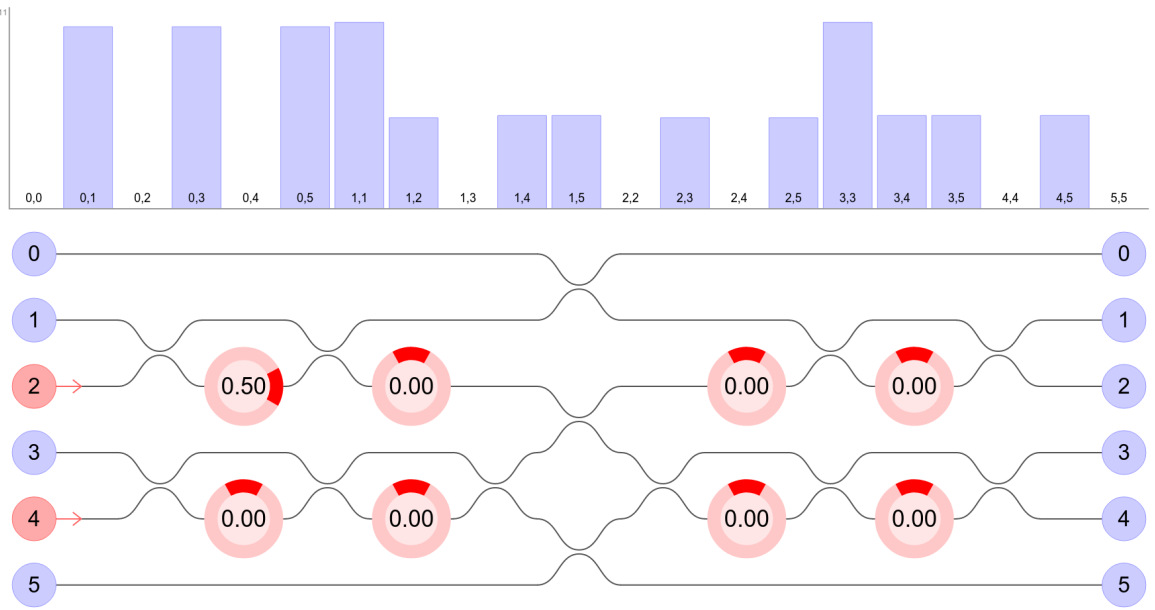


Figure : A 1-qubit+1-qubit adder can be designed using the CNOT gate and its qubit mapping $|0\rangle \rightarrow (1)$ and $|1\rangle \rightarrow (2)$ for the first qubit (control qubit) and $|0\rangle \rightarrow (3)$ and $|1\rangle \rightarrow (4)$ for the second (target qubit). A $\frac{\Pi}{2}$-phase shifter is used to mix the control qubit. After postselection, we can measure that $0+1=1$ and $1+1=0$ (outcomes (1,4) and (2,3)), carry bit is dropped.

> *If quantum mechanics hasn't profoundly shocked you, you haven't understood it yet.*
>
> Niels Bohr, *Atomic Physics and Human Knowledge*, 1958

Section 8

The future of cryptography: post-quantum cryptography

Quantum Resistant Cryptography

Currently there are 6 main different approaches:

- Lattice-based cryptography
- Multivariate cryptography
- Hash-based cryptography
- Code-based cryptography
- Supersingular Elliptic Curve Isogeny cryptography
- Symmetric Key Quantum Resistance

Annual event about PQC: PQCrypto conference (6th edition this year)

Section 9

Conclusion

Results & challenges

- Quantum computing provides a new approach to thinking & computing
- Main surprising results of the quantum mechanics theory have been verified experimentally for decades now
- A lot of progress has been made in building quantum systems suitable for computations
- Efforts are now focused on finding better qubits candidates (decoherence time), enhancing scalability of quantum chips and improving quantum error correction codes
- Absolutely nothing prevents us to increase scalabity of quantum computers
- Current asymmetric cryptosystems will probably be broken in 10 to 25 years

Bibliography

 *The age of the qubit - A new era of quantum information in science and technology*, IOP Institute of Physics, 2011.

 Jonathan P. Dowling, *Schrödinger's Killer App - Race to Build the World's First Quantum Computer*, CRC Press, 2013.

 Noson S. Yanofsky & Mirco A. Mannucci, *Quantum computing for computer scientists*, Cambridge University Press, 2008.

 Tzvetan S. Metodi & Arvin I. Faruque & Frederic T. Chong, *Quantum computing for Computer Architects*, Mark D. Hill - Series Editor, Second Edition 2011.

 Michael A. Nielsen & Isaac L. Chuang, *Quantum Computation and Quantum Information*, Cambridge University Press, 10th Anniversary Edition 2010.

Thanks for your attention!

Any questions?

✉ renaud.lifchitz@oppida.fr