Microsoft Research

Faculty Summit

2014 15TH ANNUAL

Microsoft Research Faculty Summit 2014 15TH ANNUAL

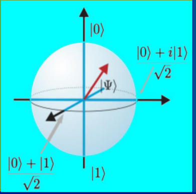
$\text{LIQ}Ui|\rangle$

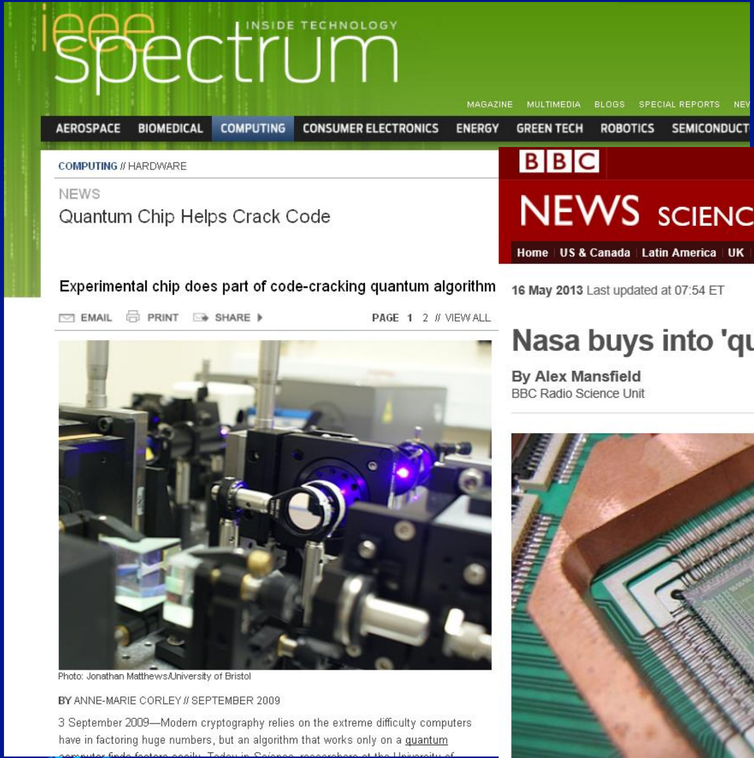
Programming a Quantum Computer

Dave Wecker
QuArC Chief Architect, Microsoft Research

Microsoft

Quantum Computing in the Media





COMPUTING // HARDWARE

NEWS

Quantum Chip Helps Crack Code

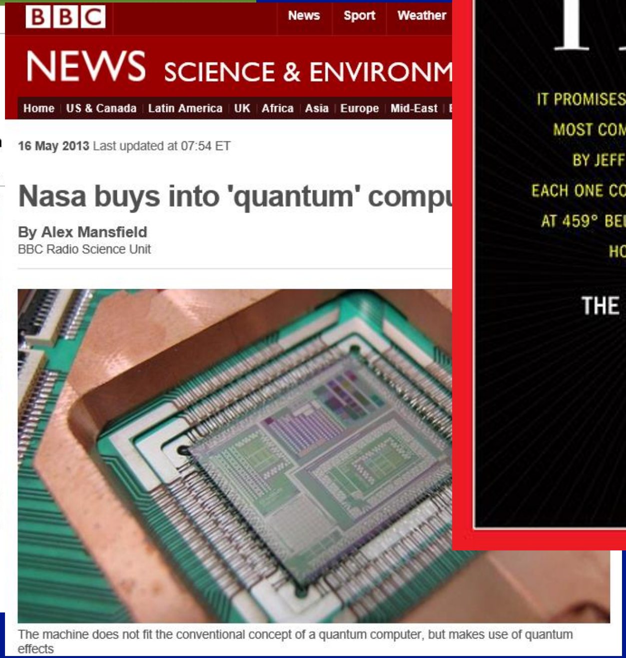
Experimental chip does part of code-cracking quantum algorithm

EMAIL PRINT SHARE PAGE 1 2 // VIEW ALL

Photo: Jonathan Matthews/University of Bristol

BY ANNE-MARIE CORLEY // SEPTEMBER 2009

3 September 2009—Modern cryptography relies on the extreme difficulty computers have in factoring huge numbers, but an algorithm that works only on a quantum

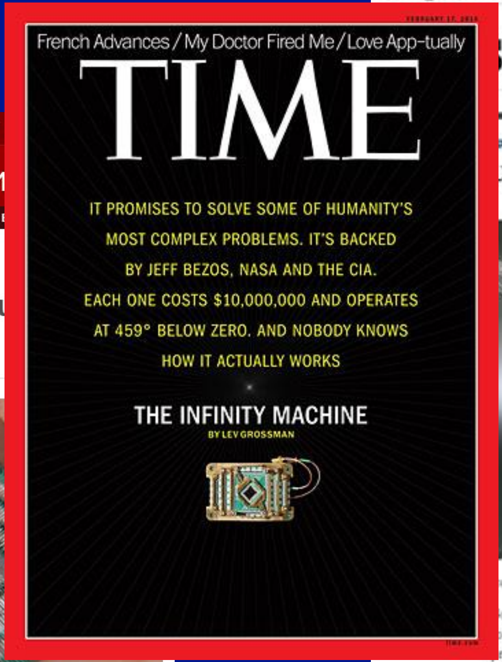


16 May 2013 Last updated at 07:54 ET

Nasa buys into 'quantum' compu

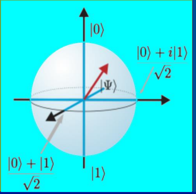
By Alex Mansfield
BBC Radio Science Unit

The machine does not fit the conventional concept of a quantum computer, but makes use of quantum effects



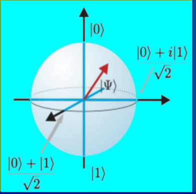
uys a Quantum Computer

oogle and a corporation associated with NASA are forming a laboratory o study artificial intelligence by means of computers that use the unusual roperties of quantum physics. Their quantum computer, which performs complex calculations thousands of times faster than existing supercomputers, is expected to be in active use in the third quarter of this



A Little Motivation

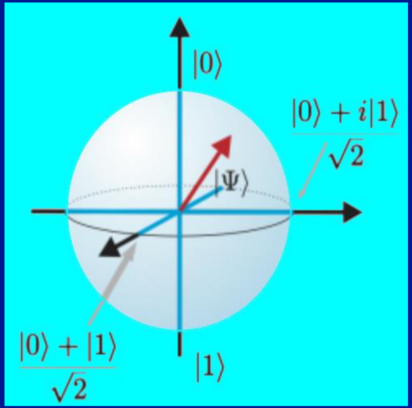
- Nitrogen Fixation:
  - Making fertilizer uses a process from 1909 and uses lots of energy (400^oC/200 atm)
  - Cost: 3-5% of the worlds natural gas production (1-2% of the world's annual energy)
  - Design of a new catalyst would take ~100-200 qubits (inexpensive fertilizer)
- Carbon Capture:
  - Cost: Capturing at point sources will result in 21-90% increase in energy cost
  - Design a new catalyst to extract CO_2 from the air would take ~200-400 qubits
- Design of new chemicals and materials:
  - Today 1/3 of all supercomputing time is spent on chemistry and materials modeling
  - Designs that can never be done classically are solvable with a few hundred qubits
  - Pharmaceuticals, High temperature Super Conductors (energy, transportation…)
    - Example: gain back current 6.5% transmission loss in power lines



Classical vs. Quantum Computing

Basic unit: **bit** = 0 or 1

A qubit lies on the surface of what is known as the Bloch sphere:

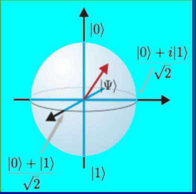


Basic unit: **qubit** = unit vector $\alpha|0\rangle + \beta|1\rangle$

- α, β are complex values ($|\alpha|^2 + |\beta|^2 = 1$)
- The Z coordinate is north-south and is our computational basis (what we can measure)

When we read a qubit we get a single bit

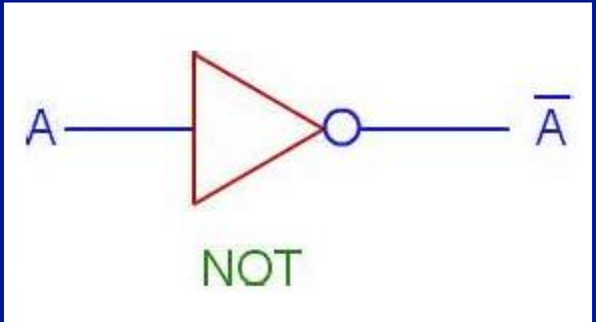
- Probability based on position along the Z axis (how close are we to $|0\rangle$ or $|1\rangle$?)
- If we're on the equator we have a 50/50 probability of measuring a 0 or a 1



Classical vs. Quantum Computing

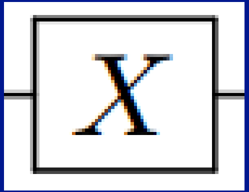
Basic unit: **bit** = 0 or 1

Computing: **logical** operation

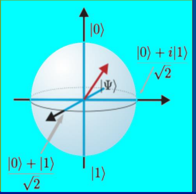


Basic unit: **qubit** = unit vector $\alpha|0\rangle + \beta|1\rangle$

Computing: **unitary** operation



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



Classical vs. Quantum Computing

Basic unit: **bit** = 0 or 1

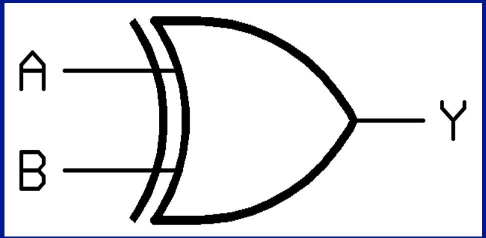
Computing: **logical** operation

Description: **truth table**

Basic unit: **qubit** = unit vector $\alpha|0\rangle + \beta|1\rangle$

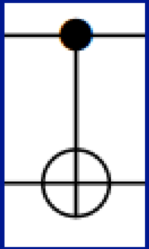
Computing: **unitary** operation

Description: **unitary matrix**



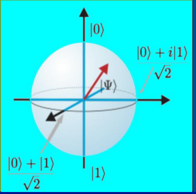
XOR gate

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0



CNOT gate

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$



Classical vs. Quantum Computing

Basic unit: **bit** = 0 or 1

Computing: **logical** operation

Description: **truth table**

Direction: Most gates only run **forward**

Copying: Independent copies are easy

Noise: Manageable with minimal ECC

Input/Output: Linear

Storage: n bits hold 1 value from 0 to $2^n - 1$

Computation:

An n-bit ALU: 1 operation/cycle

Basic unit: **qubit** = unit vector $\alpha|0\rangle + \beta|1\rangle$

Computing: **unitary** operation

Description: **unitary matrix**

Direction: Most gates are **reversible** (matrices)

Copying: Independent copies are impossible

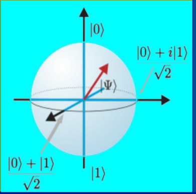
Noise: Difficult to overcome. Sophisticated QECC

Input: Linear, Output: Probabilistic (sub-linear)

Storage: n qubits can hold 2^n values

Computation:

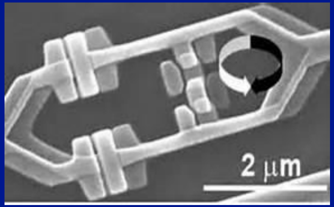
An n-qubit ALU: 2^n operations/cycle



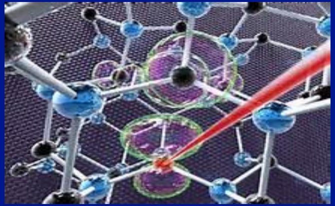
Quantum Technologies

Ion traps

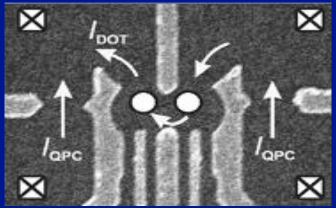
Super-conductors



NV centers

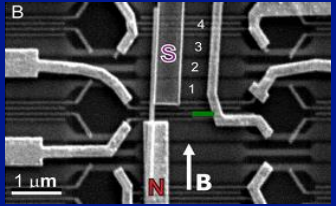


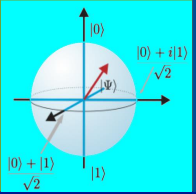
Quantum dots



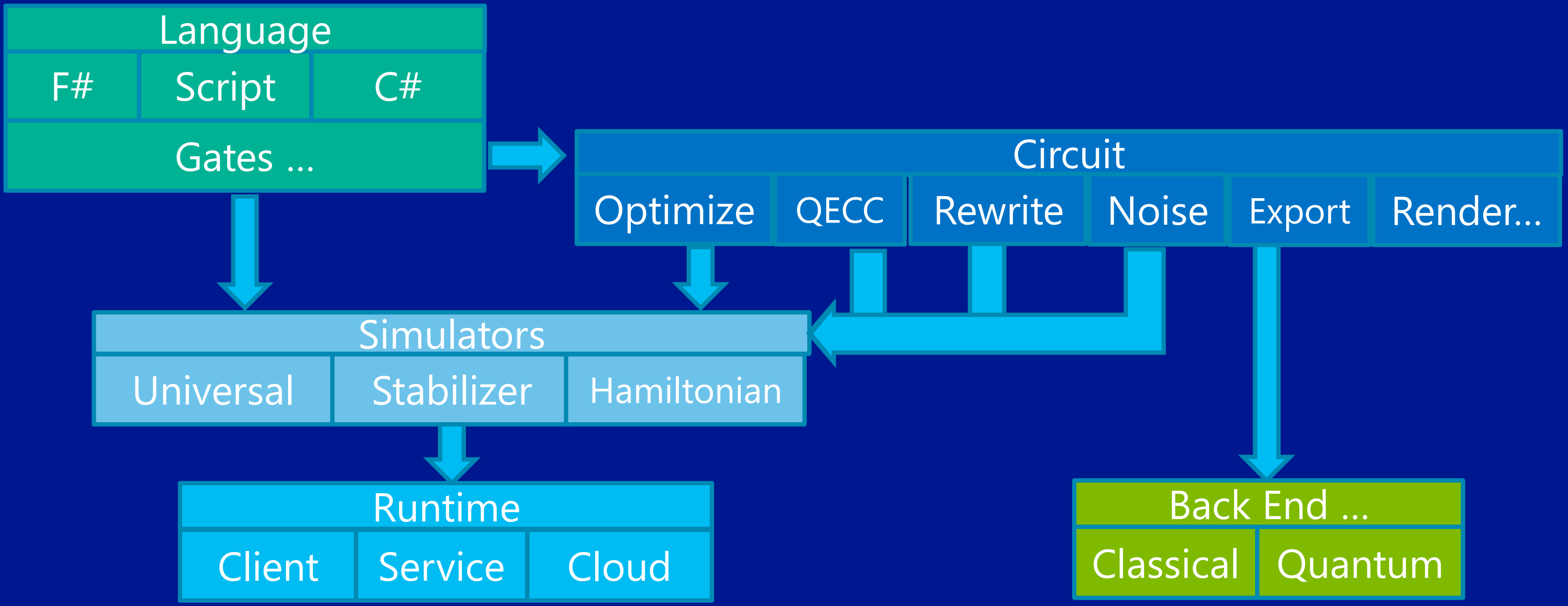
Linear optics

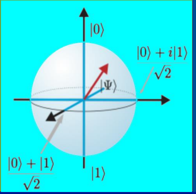
Topological





The LIQ$Ui|\rangle$ Simulation Platform



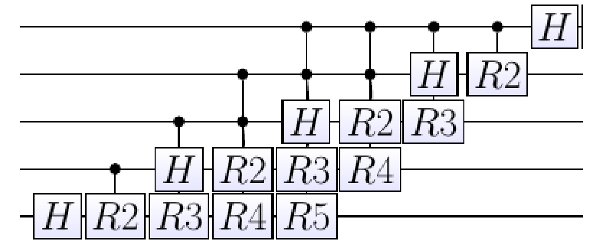


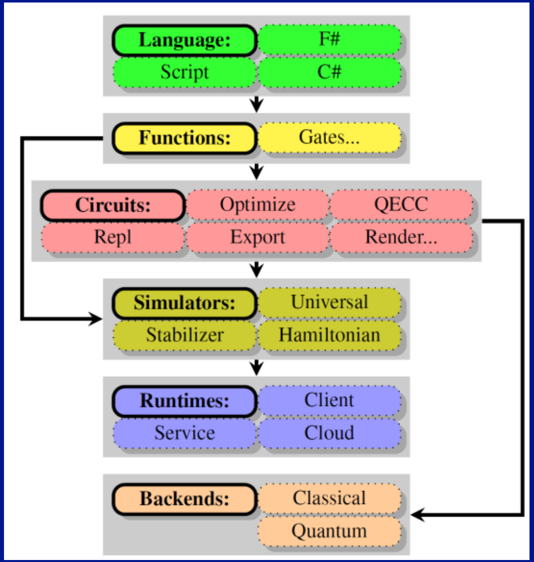
Recent paper on LIQUi|⟩

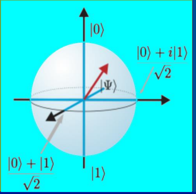
LIQUi|>: A Software Design Architecture and Domain-Specific Language for Quantum Computing. Dave Wecker, Krysta M. Svore

Languages, compilers, and computer-aided design tools will be essential for scalable quantum computing, which promises an exponential leap in our ability to execute complex tasks. LIQUi|> is a modular software architecture designed to control quantum hardware. It enables easy programming, compilation, and simulation of quantum algorithms and circuits, and is independent of a specific quantum architecture. LIQUi|> contains an embedded, domain-specific language designed for programming quantum algorithms, with F# as the host language. It also allows the extraction of a circuit data structure that can be used for optimization, rendering, or translation. The circuit can also be exported to external hardware and software environments. Two different simulation environments are available to the user which allow a trade-off between number of qubits and class of operations. LIQUi|> has been implemented on a wide range of runtimes as back-ends with a single user front-end. We describe the significant components of the design architecture and how to express any given quantum algorithm.

http://arxiv.org/abs/1402.4467

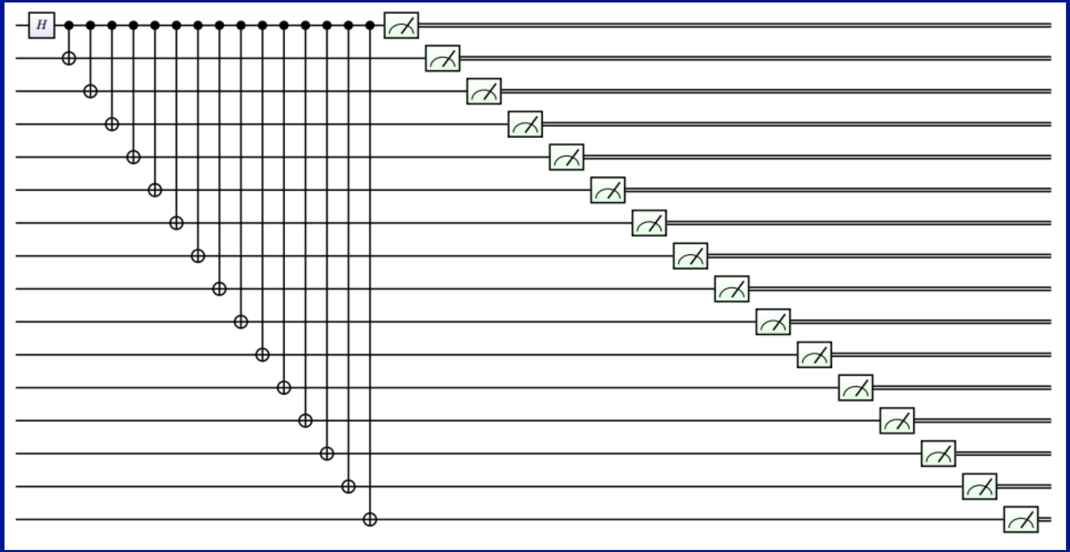




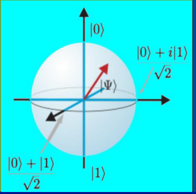


Efficient quantum implementation language

```
let entangle (qs:Qubits) =
    H qs; let q0  = qs.Head
    for q in qs.Tail do CNOT[q0;q]
    M >< qs
```

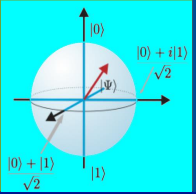



```
0:0000.0/#### Iter   0 [  0.2030]: 0000000000000
0:0000.0/#### Iter   1 [  0.1186]: 0000000000000
0:0000.0/#### Iter   2 [  0.0895]: 0000000000000
0:0000.0/#### Iter   3 [  0.0749]: 0000000000000
0:0000.0/#### Iter   4 [  0.0664]: 1111111111111
0:0000.0/#### Iter   5 [  0.0597]: 0000000000000
0:0000.0/#### Iter   6 [  0.0550]: 1111111111111
0:0000.0/#### Iter   7 [  0.0512]: 0000000000000
0:0000.0/#### Iter   8 [  0.0484]: 0000000000000
0:0000.0/#### Iter   9 [  0.0463]: 0000000000000
0:0000.0/#### Iter  10 [  0.0446]: 0000000000000
0:0000.0/#### Iter  11 [  0.0432]: 1111111111111
0:0000.0/#### Iter  12 [  0.0420]: 0000000000000
0:0000.0/#### Iter  13 [  0.0410]: 0000000000000
0:0000.0/#### Iter  14 [  0.0402]: 0000000000000
0:0000.0/#### Iter  15 [  0.0399]: 0000000000000
0:0000.0/#### Iter  16 [  0.0392]: 1111111111111
0:0000.0/#### Iter  17 [  0.0387]: 1111111111111
0:0000.0/#### Iter  18 [  0.0380]: 0000000000000
0:0000.0/#### Iter  19 [  0.0374]: 1111111111111
```

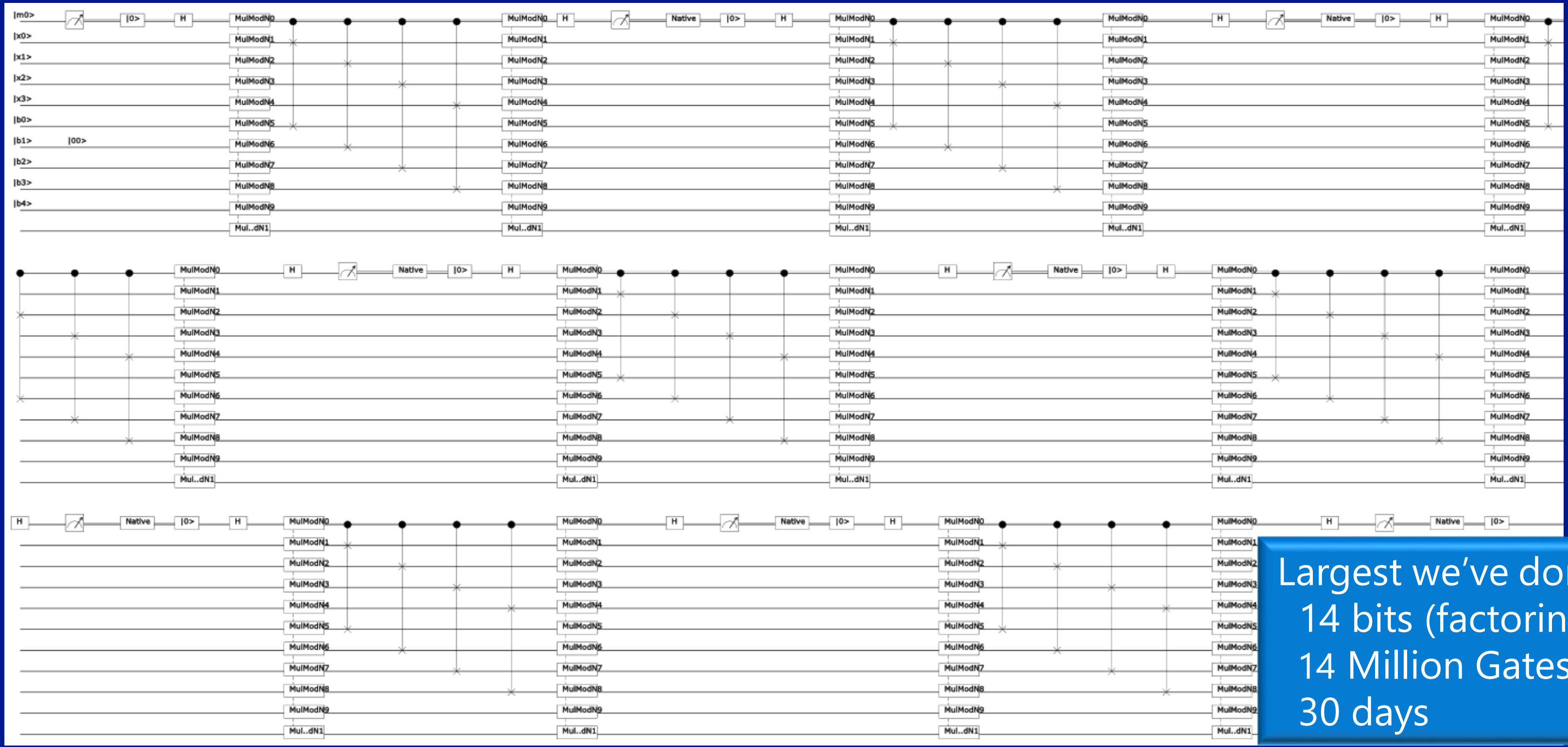


User definition of a gate

```
/// <summary>
/// Controlled NOT gate
/// </summary>
/// <param name="qs"> Use first two qubits for gate</param>
[<LQD>]
let CNOT (qs:Qubits) =
    let gate =
        Gate.Build("CNOT",fun () ->
            new Gate(
                Name    = "CNOT",
                Help    = "Controlled NOT",
                Mat     = CSMat(4,[(0,0,1.,0.);(1,1,1.,0.);
                                   (2,3,1.,0.);(3,2,1.,0.)]),
                Draw    = "\\ctrl{#1}\\go[#1]\\targ"
        ))
    gate.Run qs
```

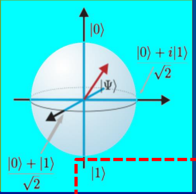


Shor's algorithm: Full Circuit:
4 bits ≅ 8200 gates

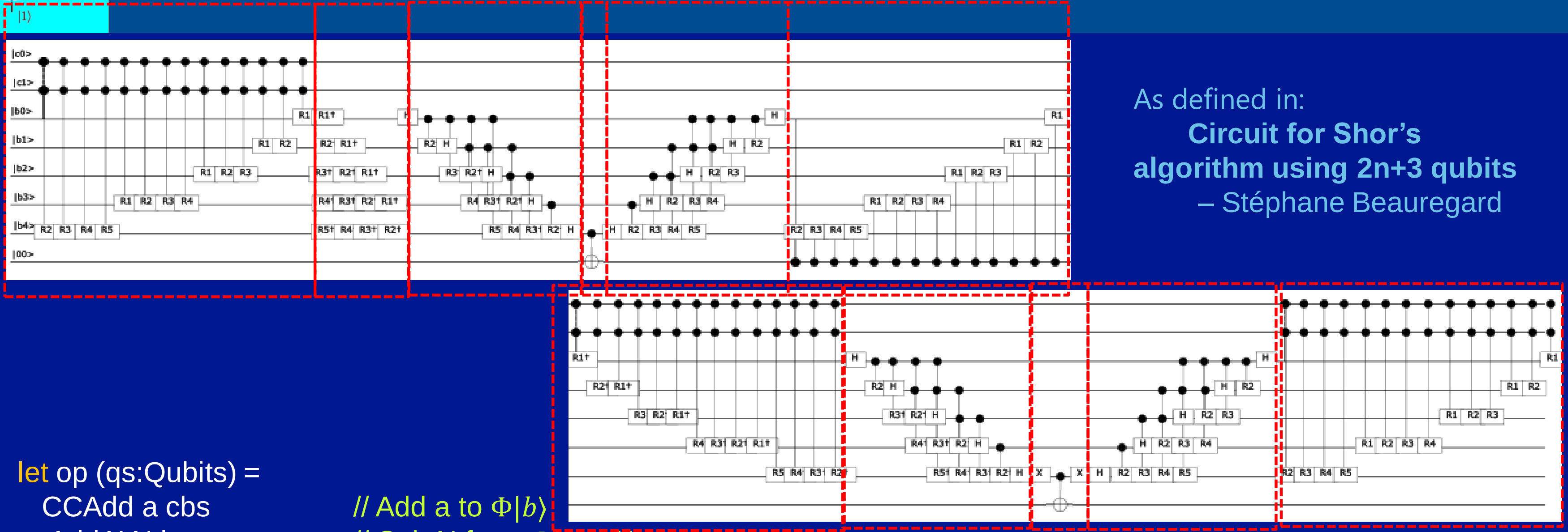


Largest we've done:
14 bits (factoring 8193)
14 Million Gates
30 days

Circuit for Shor's algorithm using 2n+3 qubits – Stéphane Beauregard

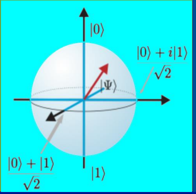


Shor's algorithm: Modular Adder

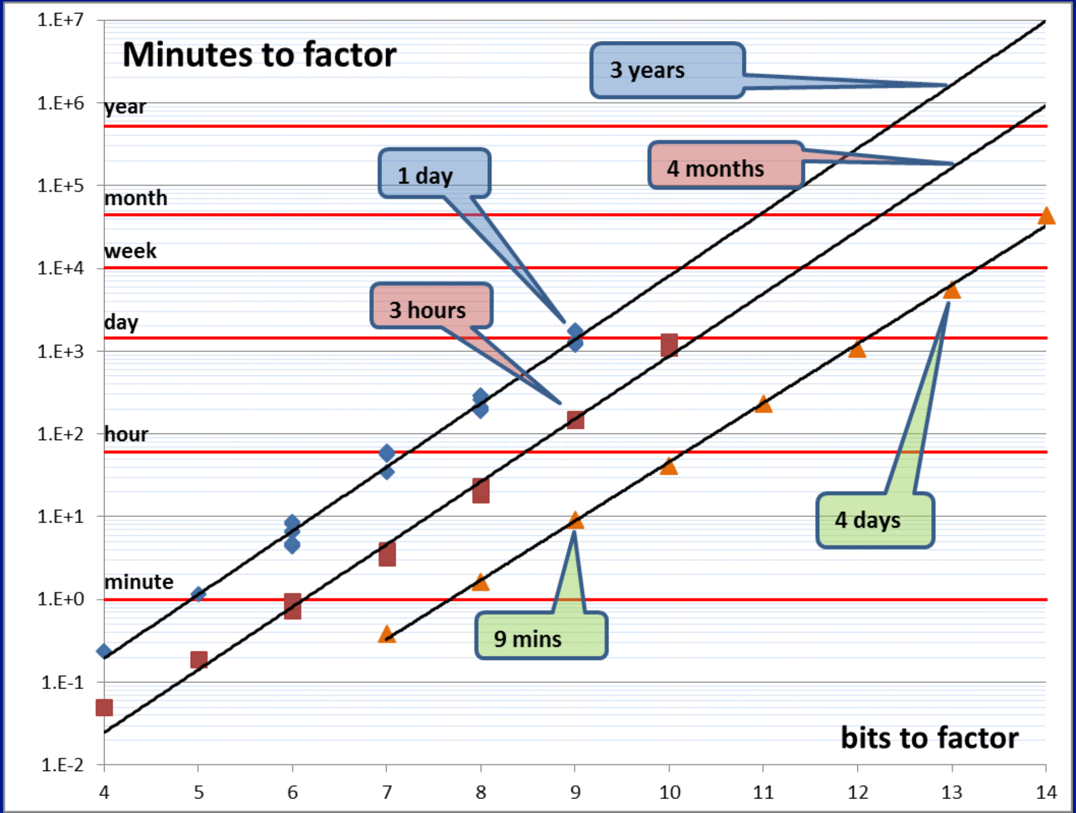


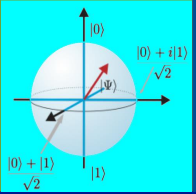
As defined in:
Circuit for Shor's algorithm using 2n+3 qubits
– Stéphane Beauregard

let op (qs:Qubits) =
CCAdd a cbs // Add a to $\Phi|b\rangle$
AddA' N bs // Sub N from $\Phi|a+b\rangle$
QFT' bs // Inverse QFT of $\Phi|a+b-N\rangle$
CNOT [bMx;anc] // Save top bit in Ancilla
QFT bs // QFT of a+b-N
CAddA N (anc :: bs) // Add back N if negative
CCAdd' a cbs // Subtract a from $\Phi|a+b\ mod\ N\rangle$
QFT' bs // Inverse QFT
X [bMx] // Flip top bit
CNOT [bMx;anc] // Reset Ancilla to $|0\rangle$
X [bMx] // Flip top bit back
QFT bs // QFT back
CCAdd a cbs // Finally get $\Phi|a+b\ mod\ N\rangle$



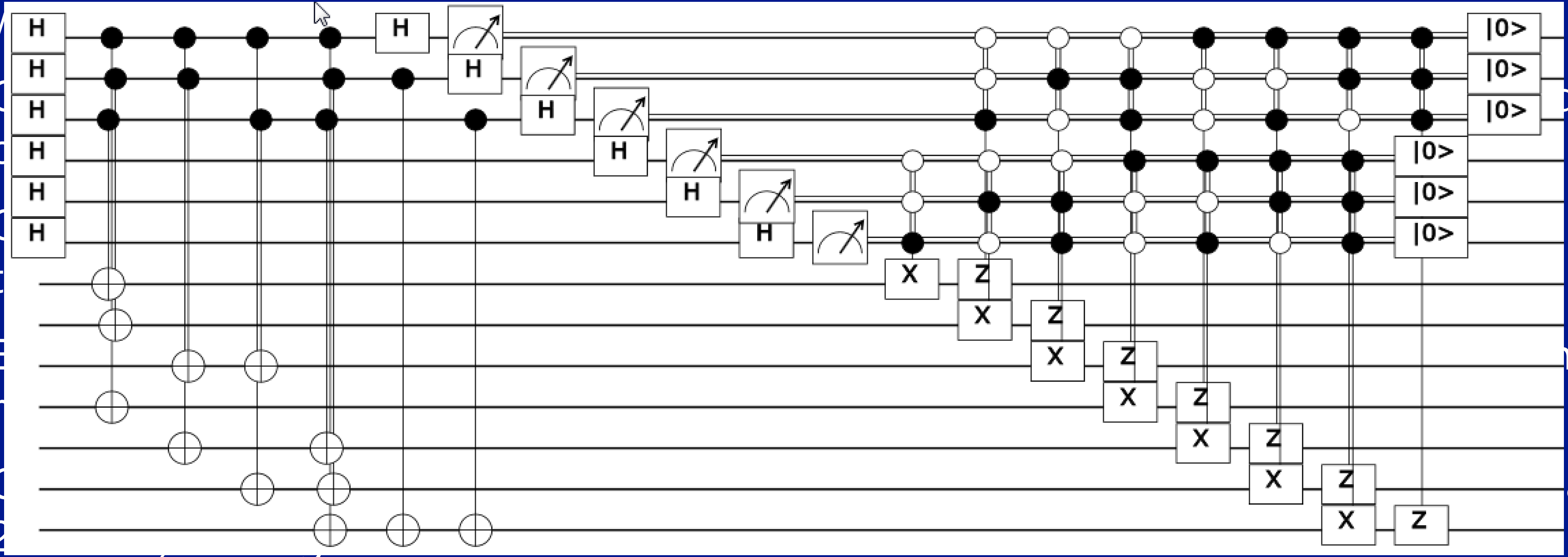
Shor's algorithm results

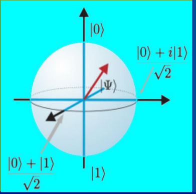




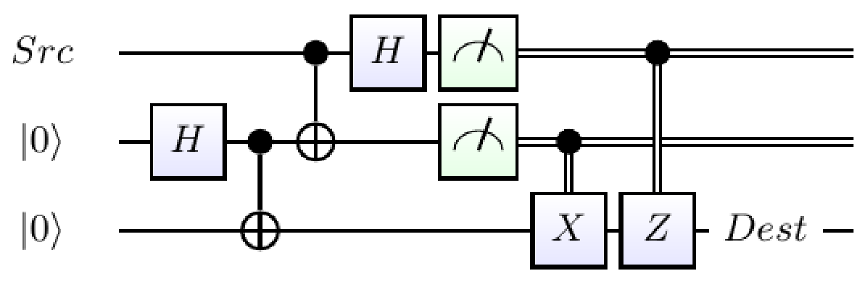
QECC: Quantum Error Correction Codes

- LIQ$Ui|\rangle$ has a user extensible module allowing circuits to be re-written automatically

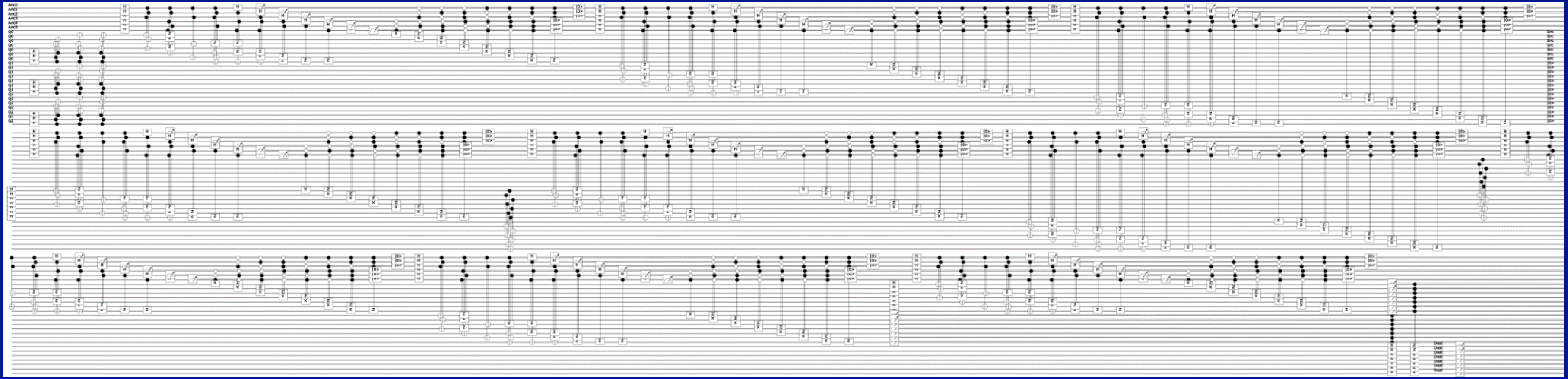


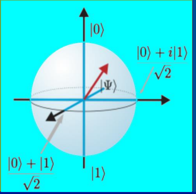


Full Teleport Circuit in a Steane7 Code



3 qubits go to 27





Spin-Glass Models

$$H(t) = \Gamma(t)\sum_{i=1}^{N}\Delta_i\sigma_i^x + \Lambda(t)\left(\sum_{i=1}^{N}h_i\sigma_i^z + \sum_{i,j=1}^{N}J_{ij}\sigma_i^z\sigma_j^z\right)$$

Quantum anneali
Sergio Boixo, Troel
David Wecker, Dan

Quantum technolo
devices, such as qu
random number ge
with capabilities ex
annealer, in particu
evolving a known i
towards the groun
problem. Here, we
qubit D-Wave One
The strong correlat
quantum annealer,
device and classica
demonstrate that t
find additional evid
small-gap avoided
problems. To asses
compare it to opti

F. Rønnow, Zhihui
vid Wecker, John

uantum devices
are the
ces, and of how to
ine and measure
avoid pitfalls that
our discussion with
ve Two device with
e device on random
ed classical and
n speedup when the
e results when
nstance basis. Our
the possibility of
that quantum
posed.

1401.2910

Yes, you can

Sciencexpress

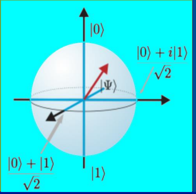
Defining and detecting quantum speedup

Troels F. Rønnow,[1] Zhihui Wang,[2,3] Joshua Job,[3,4] Sergio Boixo,[5] Sergei V. Isakov,[6] David Wecker,[7] John M. Martinis,[8] Daniel A. Lidar,[2,3,4,9] Matthias Troyer[1]*

[1]Theoretische Physik, ETH Zurich, 8093 Zurich, Switzerland. [2]Department of Chemistry, University of Southern California, Los Angeles, CA 90089, USA. [3]Center for Quantum Information Science and Technology, University of Southern California, Los Angeles, CA 90089, USA. [4]Department of Physics, University of Southern California, Los Angeles, CA 90089, USA. [5]Google, 150 Main Street, Venice Beach, CA 90291, USA. [6]Google, Brandschenkestrasse 110, 8002 Zurich, Switzerland. [7]Quantum Architectures and Computation Group, Microsoft Research, Redmond, WA 98052, USA. [8]Department of Physics, University of California Santa Barbara, CA 93106–9530, USA. [9]Department of Electrical Engineering, University of Southern California, Los Angeles, CA 90089, USA.

*Corresponding author. E-mail: troyer@phys.ethz.ch

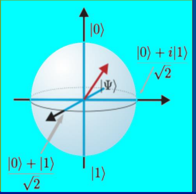
The development of small-scale quantum devices raises the question of how to fairly assess and detect quantum speedup. Here we show how to define and measure quantum speedup, and how to avoid pitfalls that might mask or fake such a speedup. We illustrate our discussion with data from tests run on a D-Wave Two device with up to 503 qubits. Using random spin glass instances as a benchmark, we find no evidence of quantum speedup when the entire data set is considered, and obtain inconclusive results when comparing subsets of instances on an instance-by-instance basis. Our results do not rule out the possibility of speedup for other classes of problems and illustrate the subtle nature of the quantum speedup question.



Quantum Walks (PageRank example)

- Start with a standard stochastic probability matrix for PageRank (G)
- Define a Hamiltonian: $\mathcal{H} = (\mathbb{I} - G)^{\dagger}(\mathbb{I} - G)$
- Convert to a Unitary: $U = e^{-i\mathcal{H}}$
- Evolve from a starting state of the static probabilities (or perform an adiabatic evolution in a 2nd quantized form)
- Accumulate average probabilities of evolving state vector
- Example: Synthetic web graph (recursive matrix definition) of 256 pages takes 8 qubits

Adiabatic quantum algorithm for search engine ranking
Garnerone, Zanardi, Lidar (arXiv.org/1109.6546)



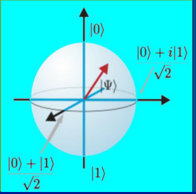
Machine Learning

Translating classical algorithms is usually not the best approach:

1. You have to load all the data (at least linear time)
2. You have to process the data (may be exponentially faster)
3. You get to read-out one number as an answer (which is probabilistic)
4. Want another answer? Go back to step 1

However….

- **Quantum algorithm for solving linear systems of equations** (http://arxiv.org/abs/0811.3171)
  - Example of building a machine learning model (efficient to build, terrible to read the model out)
- **Preconditioned quantum linear system algorithm** (http://arxiv.org/abs/1301.2340)
  - Example of asking the right question (don't ask for the model, use it)
- True exponential speed up – if you can come up with the right circuit for finding inverse eigenvalues and pick various critical parameters
- Implemented in LIQ$Ui|\rangle$. Ongoing research to do full general solutions



Quantum Chemistry

$$H = \sum_{pq} h_{pq} a_p^\dagger a_q + \frac{1}{2} \sum_{pqrs} h_{pqrs}\, a_p^\dagger a_q^\dagger a_r a_s$$

Can quantum chem **computer**: Dave We Hastings, Matthias T

As quantum computers appear fea application frequently simulating structure o computati perform qu the quantu molecule t solve exact ten-fold in the require executed is m quantum computati problems, drastic alg

um Chemistry: M. B.

rotter-Suzuki based ry on a quantum mations are hmic in the s not require any operations in the parallel or increase in order in the e error at given the Hamiltonian zuki timestep. mulation and

The Trotter Step Size Required for Accurate Quantum Simulation of Quantum Chemistry
David Poulin, M. B. Hastings, Dave Wecker, Nathan Wiebe, Andrew C. Doherty, Matthias Troyer

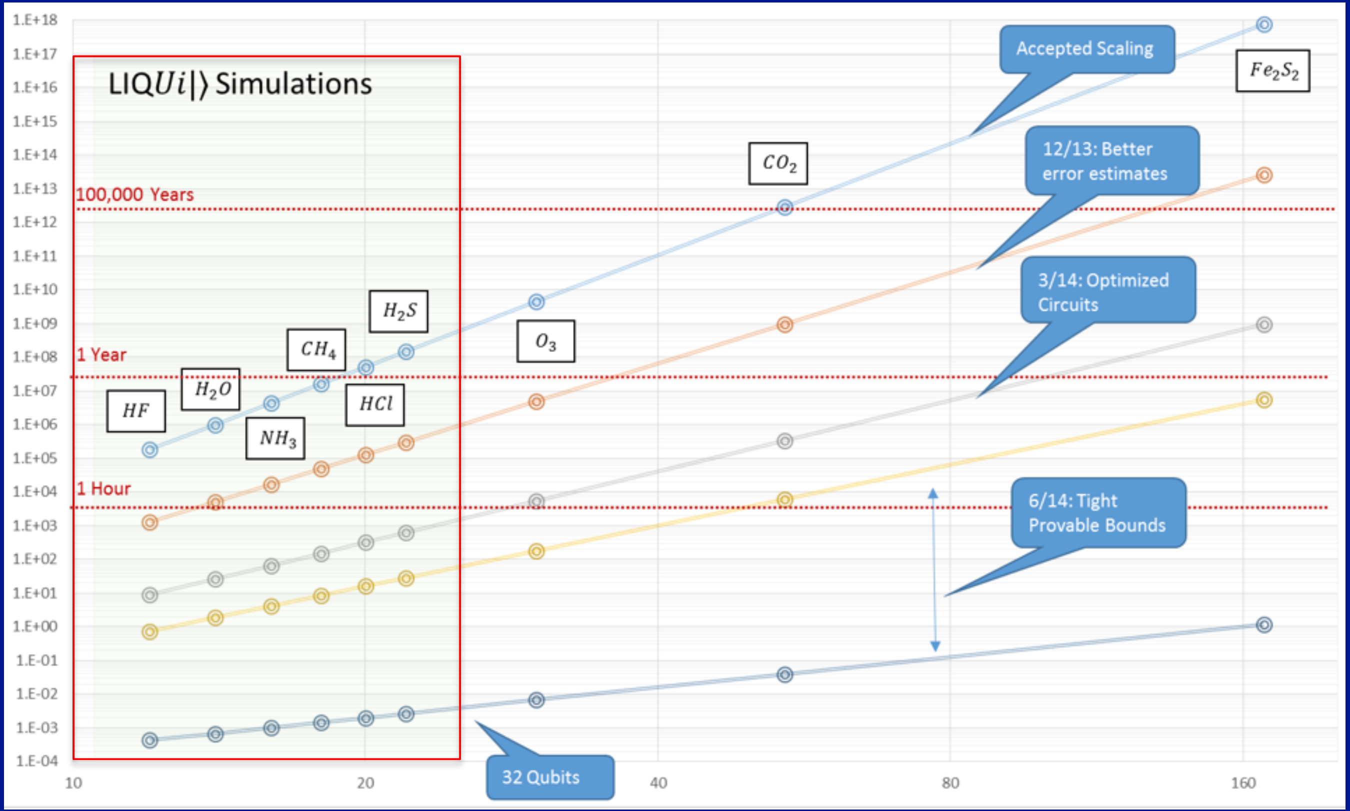
Ferredoxin (Fe_2S_2) used in many metabolic reactions including energy transport in photosynthesis

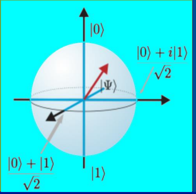
- *Intractable on a classical computer*
- *First paper: ~300 million years to solve*
- *Second paper: ~30 years to solve (10^7 reduction)*
- *Third paper: ~300 seconds to solve (another 10^3 reduction)*

http://arxiv.org/abs/1406.4920

Quantum Chemistry

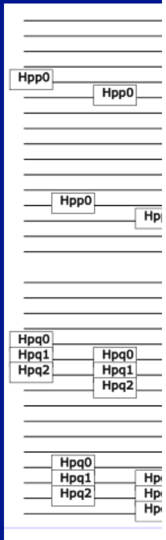
$$H = \sum_{pq} h_{pq} a_p^\dagger a_q + \frac{1}{2} \sum_{pqrs} h_{pqrs} a_p^\dagger a_q^\dagger a_r a_s$$





Designing High Temperature Superconductors

$$H = -\sum_{\langle i,j\rangle}\sum_{\sigma} t_{ij}\left(c^{\dagger}_{i,\sigma}c_{j,\sigma} + c^{\dagger}_{j,\sigma}c_{i,\sigma}\right) + U\sum_{i} n_{i,\uparrow}n_{i,\downarrow} + \sum_{i}\epsilon_i\eta_i$$



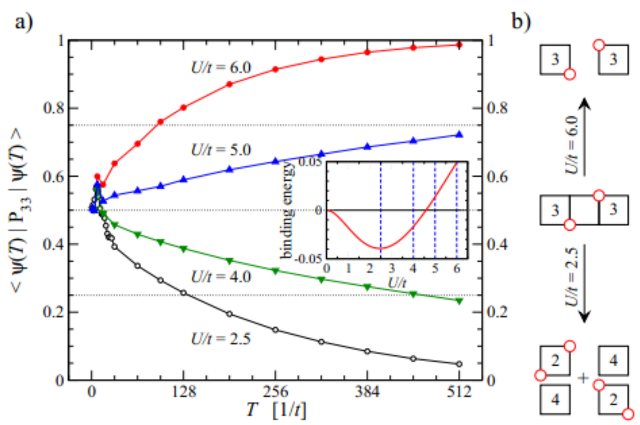
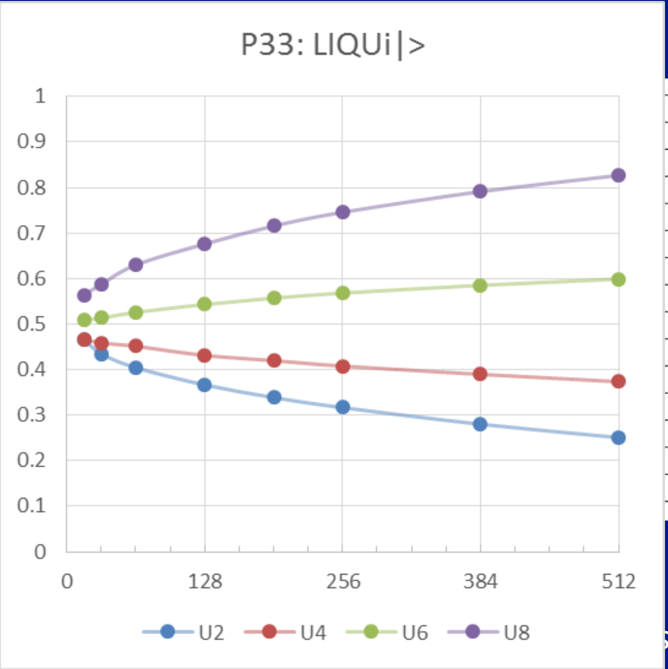
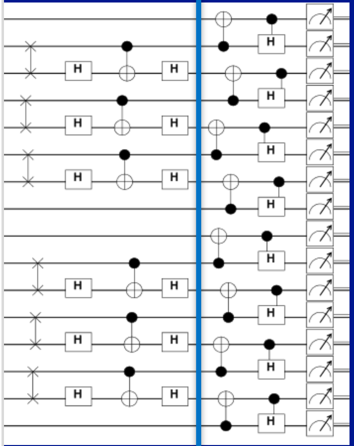


FIG. 3: Signature for d-wave RVB pairing when decoupling two plaquettes with 6 atoms for varying onsite repulsion U/t. b) For $U/t \lesssim 4.5$ pairs are formed with a small binding energy shown in the inset. In the final state after sufficiently slow decoupling the hole pair is located on one plaquette, while for large repulsion $U/t \gtrsim 4.5$ the unpaired holes separate into 3 atoms on each plaquette (right panel). a) Projection of the final state onto the subspace with 3 atoms on each plaquette.



Basis Change

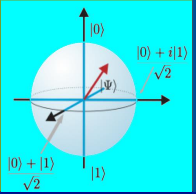


surement

See: d-wave resonating valence bond states of fermionic atoms in optical lattices

15TH ANNUAL Microsoft Research Faculty Summit 2014

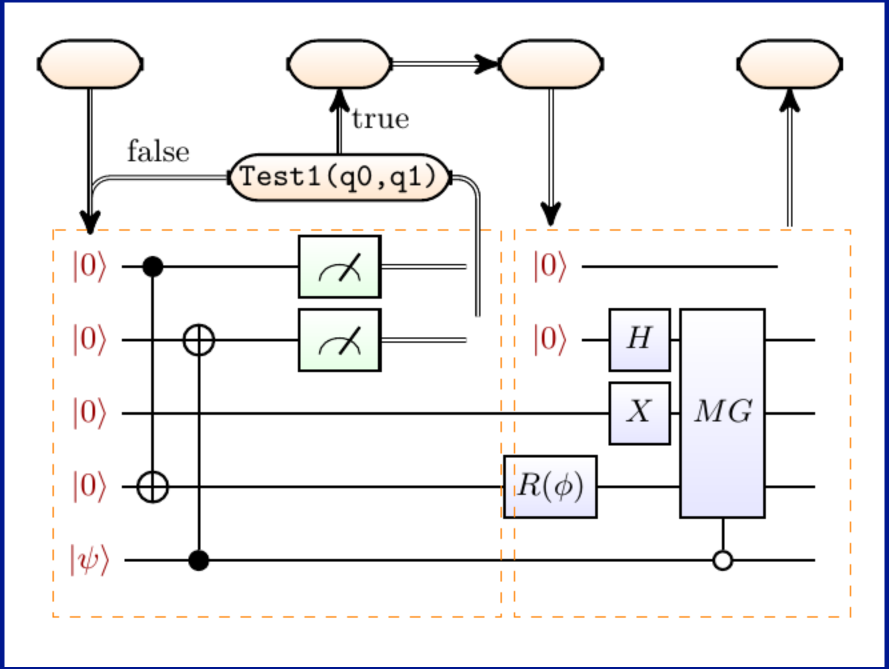
Microsoft

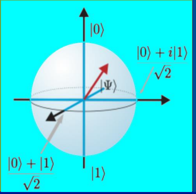


LIQ$Ui|\rangle$ for Compilation onto Hardware

```
let QFT (qs : Qs) =
    let n = qs.Length - 1
    for i = 0 to n do
        let q = qs.[i]
        H q
        for j = (i + 1) to n do
            let theta = 2.0 * Math.PI /
                float(1 <<< (j - i + 1))
            CRz theta qs.[j] q
    for i = 0 to ((n - 1) / 2) do
        SWAP qs.[i] qs.[n - i]

let QftOp  = compile QFT
let QftOp' = adjoint QftOp
```





Thank You

Dave Wecker
QuArC Chief Architect
Microsoft Corporation

Referenced papers may be found at:

http://research.microsoft.com/QuArC
http://arxiv.org/find/all/1/wecker_d

Save the planet and return your name badge before you leave (on Tuesday)

Microsoft