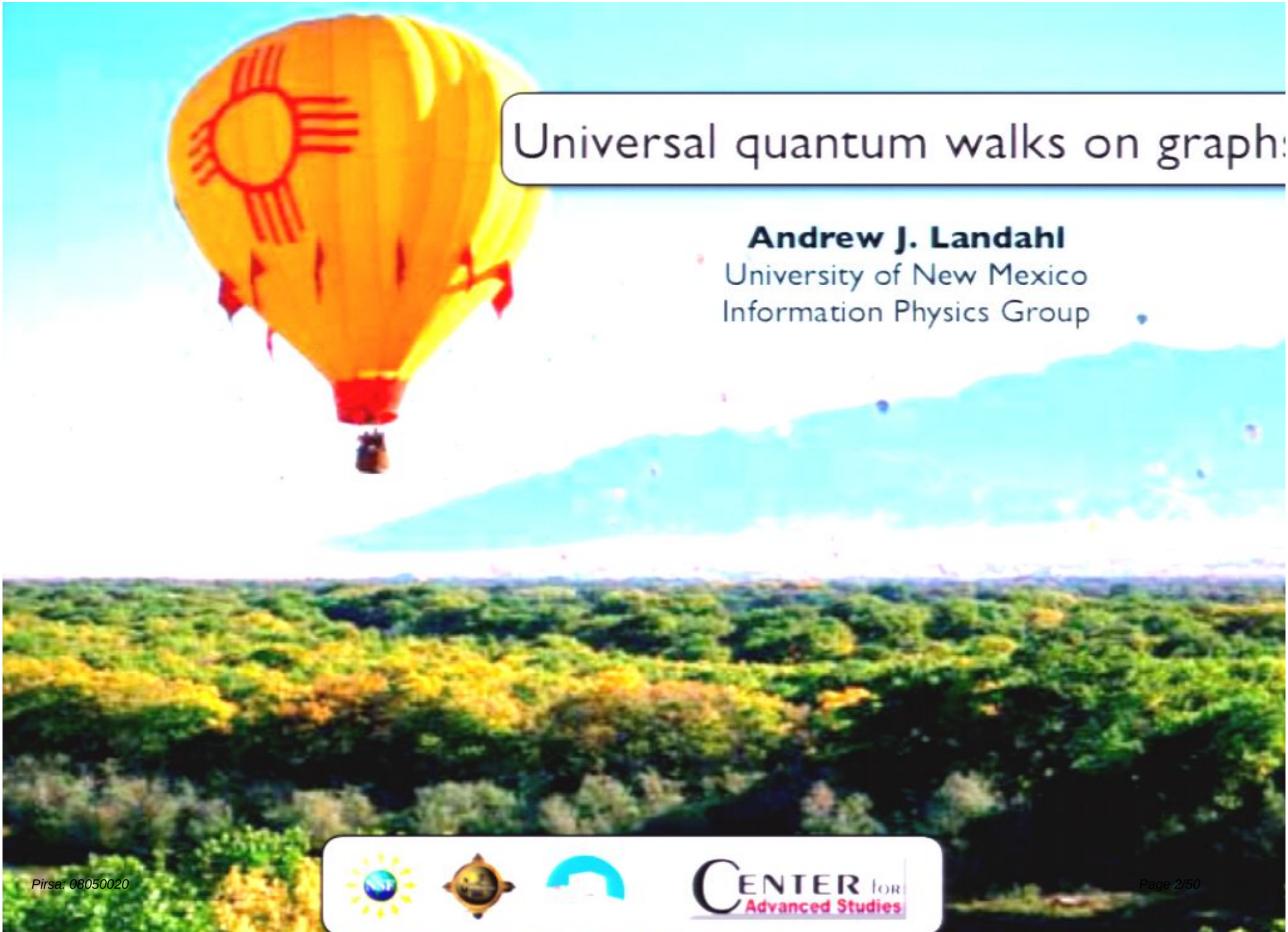
Title: Universal quantum walks on graphs

Date: May 01, 2008 09:50 AM

URL: http://pirsa.org/08050020

Abstract: We construct a family of time-independent nearest-neighbor Hamiltonians coupling eight-state systems on a 1D ring that enables universal quantum computation. Hamiltonians in this family can achieve universality either by driving a continuous-time quantum walk or by terminating an adiabatic algorithm. In either case, the universality property can be understood as arising from an efficient simulation of a programmable quantum circuit. Using gadget perturbation theory, one can demonstrate the same kind of universality for related Hamiltonian families acting on qubits in 2D. Our results demonstrate that simulating 1D chains of spin-7/2 particles is BQP-hard, and indeed BQP-complete because the outputs of decision problems can be encoded in the outputs of such simulations.


Universal quantum walks on graphs
Andrew J. Landahl
University of New Mexico
Information Physics Group
CENTER for Advanced Studies

Last talk: Quantum communication by spin chains

A B

This talk: Quantum *computation* by spin chains

Universal quantum walks and adiabatic algorithms by 1D Hamiltonians, B.A. Chase. **AJL**. arXiv:0802.1207 (2008).

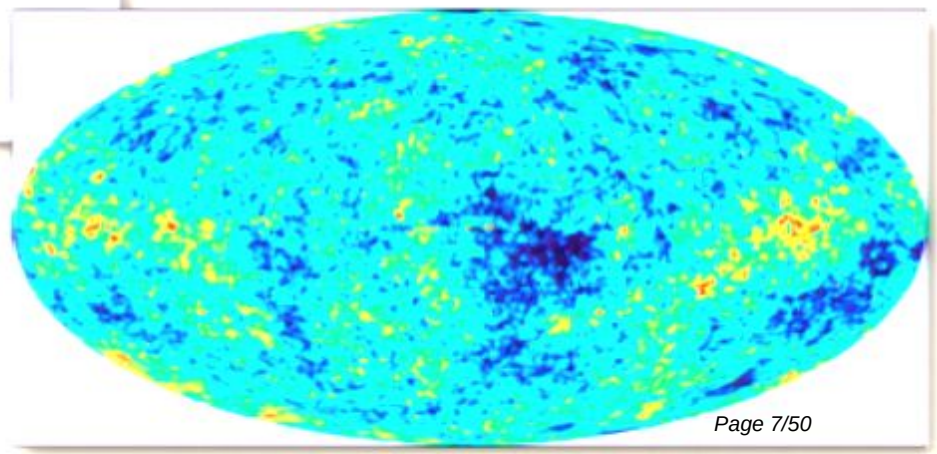
Brad Chase
UNM graduate student

A poll.

A poll.

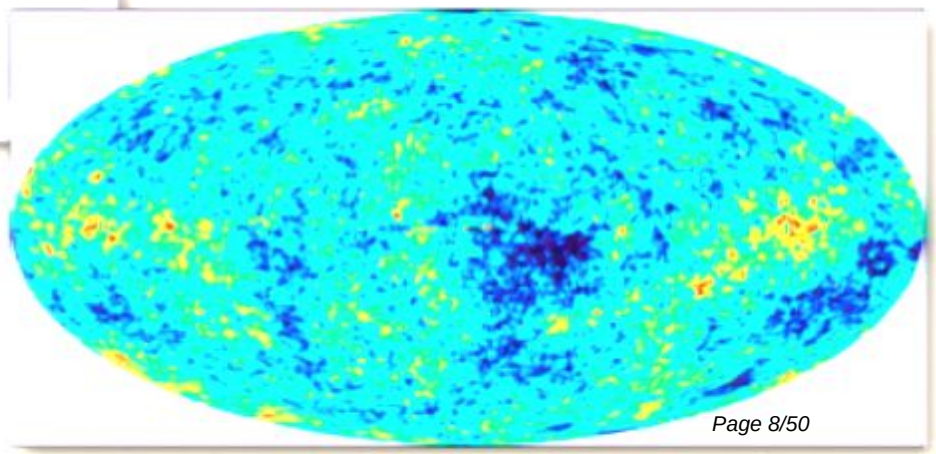
A poll.

A poll.



A poll.

1984

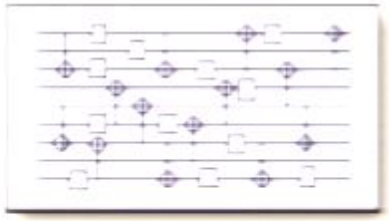


Our contribution:

Quantum walks driven by NN interactions on chains of 8-state spins are universal.

Why a spin-chain quantum computer?

Fabricating the spin-chain computer:



$U = U_T \cdots U_1$

Time-independent Hamiltonian

$H = H(U_1, \ldots, U_T)$

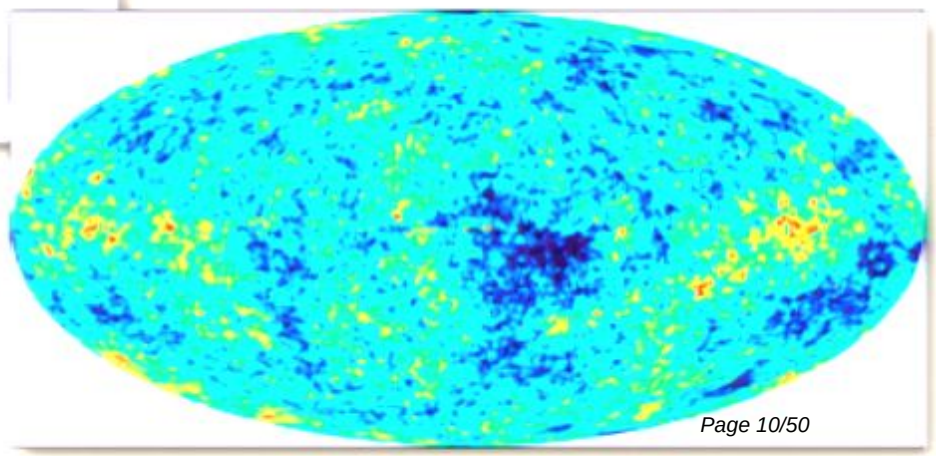
Operating the spin-chain computer:

1. *Prepare input.*
2. *Wait.*
3. *Read output.*

No need for a classical controller during the course of a computation!

A poll.

1984

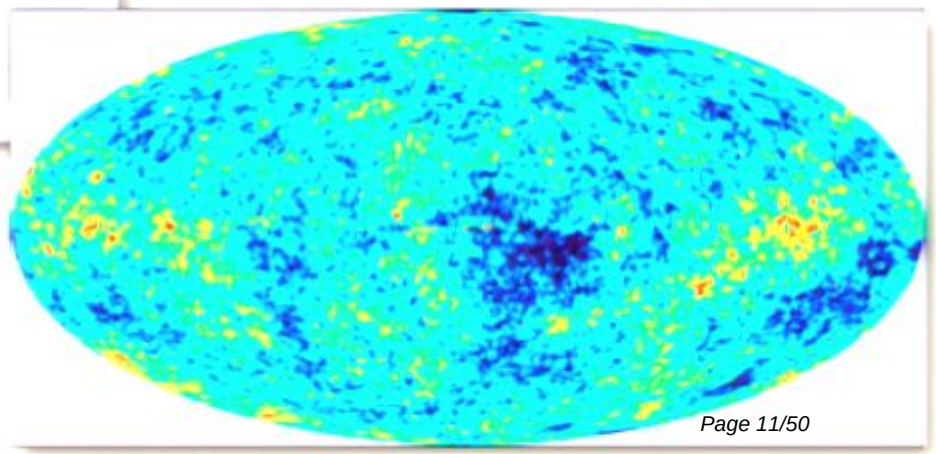


Our contribution:

Quantum walks driven by NN interactions on chains of 8-state spins are universal.

A poll.

1984

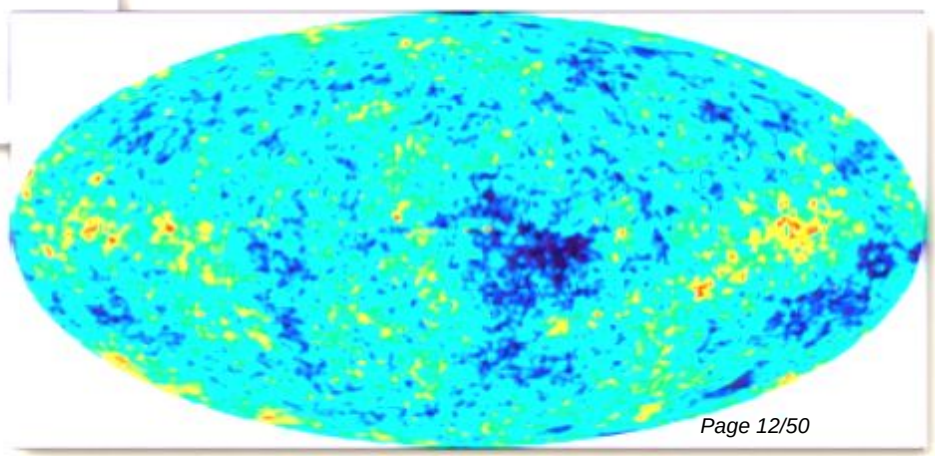


Our contribution:

Quantum walks driven by NN interactions on chains of 8-state spins are universal.

A poll.

1984

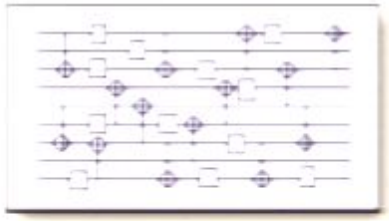


Our contribution:

Quantum walks driven by NN interactions on chains of 8-state spins are universal.

Why a spin-chain quantum computer?

Fabricating the spin-chain computer:



$U = U_T \cdots U_1$

Time-independent Hamiltonian

$H = H(U_1, \ldots, U_T)$

Operating the spin-chain computer:

1. *Prepare input.*
2. *Wait.*
3. *Read output.*

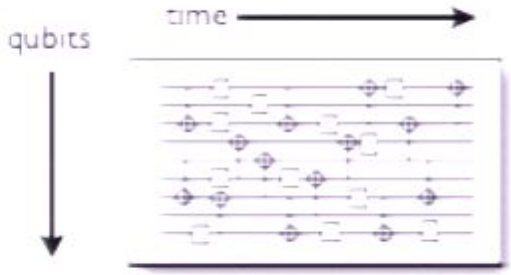
No need for a classical controller during the course of a computation!

Quantum circuit variants

Quantum circuit model:

1. n qubits (complexity parameter)
2. State set, preparation sequence
3. Gate set, gate sequence
4. Measurement set, observation sequence

} Efficiently (Turing-)computable.



Standard circuit model: New computation, new gate sequence.

Measurement-based circuit model: New computation, new measurement sequence.

Programmable circuit model: New computation, new preparation sequence.

Universality:

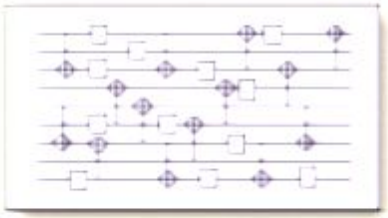
$$\|U_T \cdots U_1) - U\| < \varepsilon$$

Efficiency:

$$T = \mathrm{poly}(n, 1/\varepsilon)$$

Solovay-Kitaev algorithm: $T = \mathrm{polylog}(1/\varepsilon)$

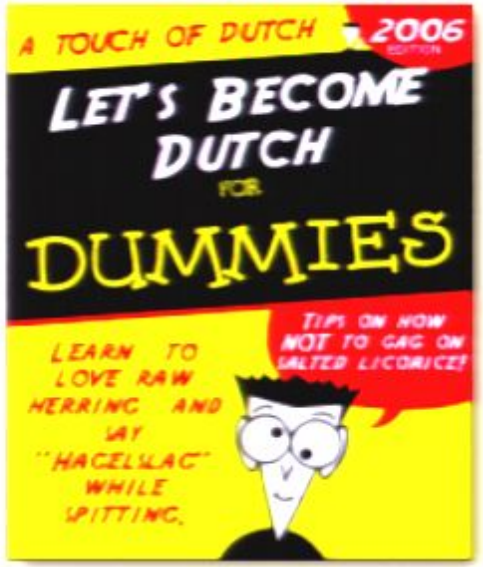
Simuleebare



$$U = U_T \cdots U_1$$

Time-independent Hamiltonian

$$H = H(U_1, \ldots, U_T)$$



Program **Data**

Universality:

$$\| e^{-iHt} - U \| < \varepsilon$$

Standard circuit: New circuit, new Hamiltonian.

Programmable circuit: New circuit, new preparation

Efficiency:

$$t = \mathrm{poly}(n, 1/\varepsilon)$$

Quantum walks

Goal: Design a programmably universal time-independent nearest-neighbor spin-chain Hamiltonian.

Evolution by a time-independent Hamiltonian:

Continuous-time quantum walk on a graph.

Continuous-time random walk:

$$\frac{d\vec{p}}{dt} = -K\vec{p}$$

Continuous-time quantum walk:

$$\frac{d|\psi\rangle}{dt} = -iH|\psi\rangle$$

Probability conservation:

$$\frac{d\|\vec{p}\|_1}{dt} = 0$$

Amplitude conservation:

$$\frac{d\||\psi\rangle\|_2}{dt} = 0$$

$$K_{ij} \in \mathbb{R} \qquad \sum_i K_{ij} = 0$$

$$H_{ij} \in \mathbb{C} \qquad H_{ij} = H_{ji}^*$$

Quantum walks

Given Hamiltonian H, construct graph G:

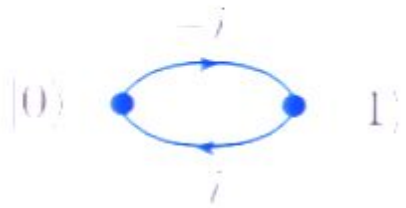
Vertices: Basis of system states.

(Weighted) Edges: Matrix elements of H in this basis.

N.B. n-qubit spin chain has 2^n states.

Example: $H = Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}$

Z-basis



Y-basis

Quantum walks

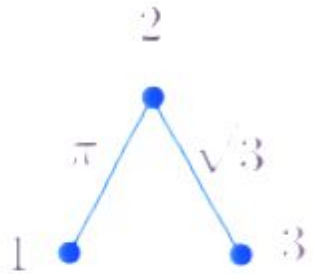
Given graph G, construct Hamiltonian H:

Row/column labels: Vertices of G.

Matrix elements: (Weights of) edges that connect vertices.

Cannot have a quantum walk on a directed graph.

Example:



$$H = \begin{pmatrix} 0 & \pi & 0 \\ \pi & 0 & \sqrt{3} \\ 0 & \sqrt{3} & 0 \end{pmatrix}$$

A continuous-time quantum walk computer

Foundations of Physics, Vol. 16, No. 6, 1986

Quantum Mechanical Computers[1]

Richard P. Feynman[2]

Received March 15, 1985

The physical limitations, due to quantum mechanics, on the functioning of computers are analyzed.

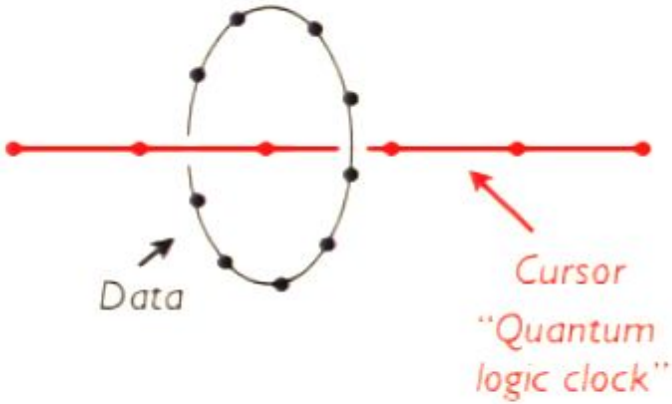
[1] *Editor's note.* This article, which is based on the author's plenary talk presented at the CLEO/IQEC Meeting in 1984, originally appeared in the February 1985 issue of *Optics News*. It is here reprinted with kind permission of Professor Feynman and *Optics News*.

[2] Department of Physics, California Institute of Technology, Pasadena, California 91125.

Standard quantum circuit: $U_T \cdots U_2 U_1$

Feynman Hamiltonian:

$$H_F = \sum_{t=1}^{T} \left(U_t \sigma_t^+ \sigma_{t-1}^- + U_t^\dagger \sigma_{t-1}^+ \sigma_t^- \right)$$



Symmetry in Feynman's Hamiltonian

$$H_F = \sum_{t=1}^{T} U_t \sigma_t^x \sigma_{t+1}^x + U_t^\dagger \sigma_{t+1}^x \sigma_t^x$$

H_F preserves the number of 1s in the cursor state:

$$H_F^{(k)} = \sum_{t=1}^{T} U_t |t\rangle\langle t-1| + U_t^\dagger |t-1\rangle\langle t|$$

$H_F^{(k)}$ is a hopping Hamiltonian on a different set of states:

$$H_F^{(k)} = \sum_{t=1}^{T} U_t |t\rangle\langle t-1| + U_t^\dagger |t-1\rangle\langle t| \qquad |c_t\rangle = \frac{1}{\sqrt{T+1}} \sum_{t=0}^{T} U_t \ldots U_1 |\psi_0\rangle |t\rangle$$

$H_F^{(k)}$ is a quantum walk on a line; can modify interactions to get perfect-fidelity state transfer:

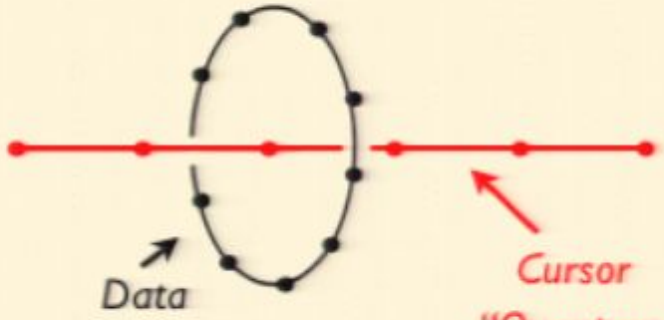
$$H_F^{(k)} = \sum_{t=1}^{T} \sqrt{t(T+1-t)}\,(U_t |t\rangle\langle t-1| + U_t^\dagger |t-1\rangle\langle t|)$$

Perfect transfer of arbitrary states in quantum spin networks
M. Christandl, N. Datta, T.C. Dorlas, A. Ekert, A. Kay, **AJL**, Phys. Rev. A **71**, 032312 (2005).

Perfect state transfer in quantum spin networks
M. Christandl, N. Datta, A. Ekert, **AJL**, Phys. Rev. Lett. **92**, 187902 (2004).

Trouble in Graphland

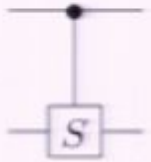
$$H_F^{eff} = \sum_{t=1}^{T} \sqrt{T(T-t)}\,(U_t \sigma_t^+ \sigma_{t-1}^- + U_t^\dagger \sigma_{t-1}^- \sigma_t^+)$$



$S = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}$ Phase —[S]—

$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$ Hadamard —[H]—

$$\Lambda(S) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & i \end{pmatrix}$$ Controlled-Phase



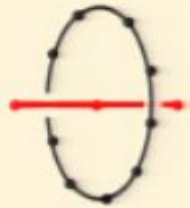
Theorem (Kitaev, 1997): $\{H, \Lambda(S)\}$ *form a universal gate basis in the standard circuit model.*

Problems:

- Need 4-body terms for universal quantum circuits
- Interactions are not spatially local

Key ideas of our solution

- $\{H, \Lambda(S), SWAP\}$ forms a nearest-neighbor universal gate set.



- Feynman's cursor is actually a *program region*.

- Cursor can be divided into constant-sized *master* program and *slave* sub-programs

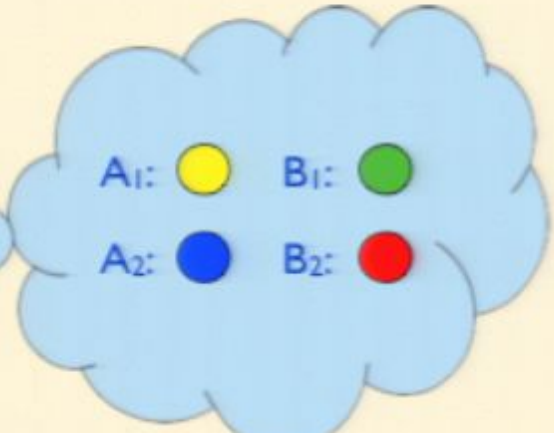
- Master program can be carried by each spin.

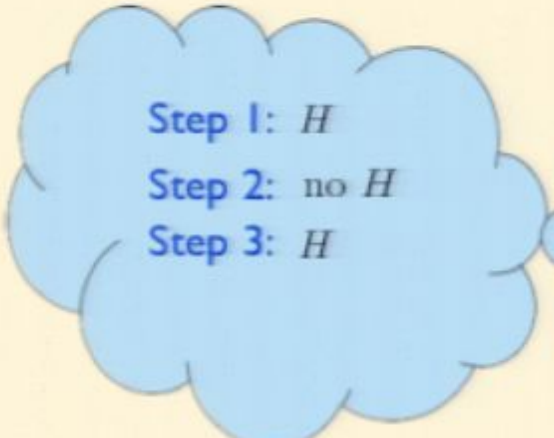
Programmable programs

Master "cursor" program

Data qubits B A

SWAP program





Hadamard progra

Programmable programs

Move the next qubit A into position!

Master "cursor" program

Data qubits B A

A_1: B_1:

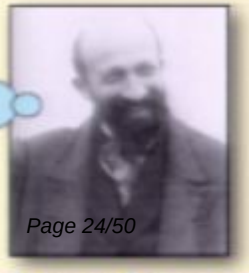
A_2: B_2:

SWAP program

Step 1: H

Step 2: no H

Step 3: H



Hadamard progra

Programmable programs

Move the next qubit A into position!

Master "cursor" program

Data qubits B A

A_1: B_1:

A_2: B_2:

SWAP program

Step 1: H

Step 2: no H

Step 3: H

Hadamard progra

Programmable programs

Move the next qubit A into position!

Master "cursor" program

Data qubits B A

Qubit A is in position!

A_1: B_1:

A_2: B_2:

SWAP program

Step 1: H

Step 2: no H

Step 3: H

Hadamard progra

Programmable programs

Move the next qubit B into position!

Master "cursor" program

Data qubits B A

A_1: B_1:

A_2: B_2:

SWAP program

Step 1: H

Step 2: no H

Step 3: H

Hadamard progra

Programmable programs

Move the next qubit B into position!

Master "cursor" program

Data qubits B A

SWAP program

A_1: B_1:

A_2: B_2:

Step 1: H

Step 2: no H

Step 3: H

Hadamard progra

Programmable programs

Move the next qubit B into position!

Master "cursor" program

Data qubits B A

A_1: B_1:

A_2: B_2:

SWAP program

Step 1: H

Step 2: no H

Step 3: H

Hadamard progra

Programmable programs

Move the next qubit B into position!

Master "cursor" program

Data qubits B A

A_1: B_1:

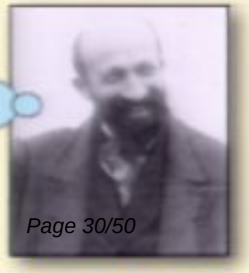
A_2: B_2:

SWAP program

Step 1: H

Step 2: no H

Step 3: H



Hadamard progra

Programmable programs

Move the next qubit B into position!

Master "cursor" program

Data qubits B A

Qubit B is in position!

A_1: B_1:

A_2: B_2:

SWAP program

Step 1: H

Step 2: no H

Step 3: H

Hadamard progra

Programmable programs

Apply $\Lambda(S)$ gate between qubits A and B, and apply H gate to qubit B if needed!

Master "cursor" program

Qubit B is in position!

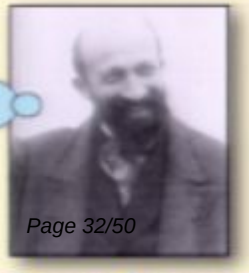
A_1: B_1:

A_2: B_2:

Step 1: H

Step 2: no H

Step 3: H



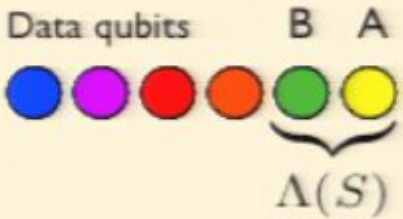
SWAP program

Hadamard progra

Programmable programs

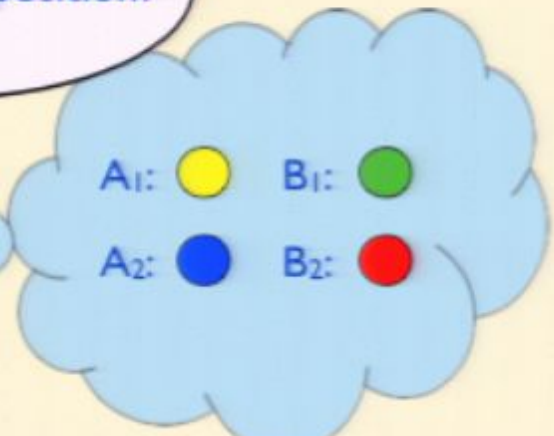
Apply $\Lambda(S)$ gate between qubits A and B, and apply H gate to qubit B if needed!

Master "cursor" program



Qubit B is in position!

SWAP program



Step 1: H
Step 2: no H
Step 3: H

Hadamard progra

Programmable programs

Apply $\Lambda(S)$ gate between qubits A and B, and apply H gate to qubit B if needed!

Master "cursor" program

Data qubits B A

Qubit B is in position!

SWAP program

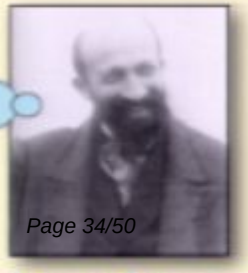
A_1: B_1:

A_2: B_2:

Step 1: H

Step 2: no H

Step 3: H



Hadamard progra

Programmable programs

Apply $\Lambda(S)$ gate between qubits A and B, and apply H gate to qubit B if needed!

Master "cursor" program

Qubit B is in position!

A_1: B_1:

A_2: B_2:

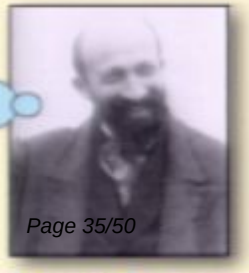
SWAP program

Done!

Step 1: H

Step 2: no H

Step 3: H



Hadamard progra

Programmable programs

Non-local program

Apply $\Lambda(S)$ gate between qubits A and B, and apply H gate to qubit B if needed!

Only three states!

Master "cursor" program

Data qubits B A

Local program

Qubit B is in position!

$\Lambda(S)^4 = I$

Local program

Done!

A_1: B_1:

A_2: B_2:

Step 1: H

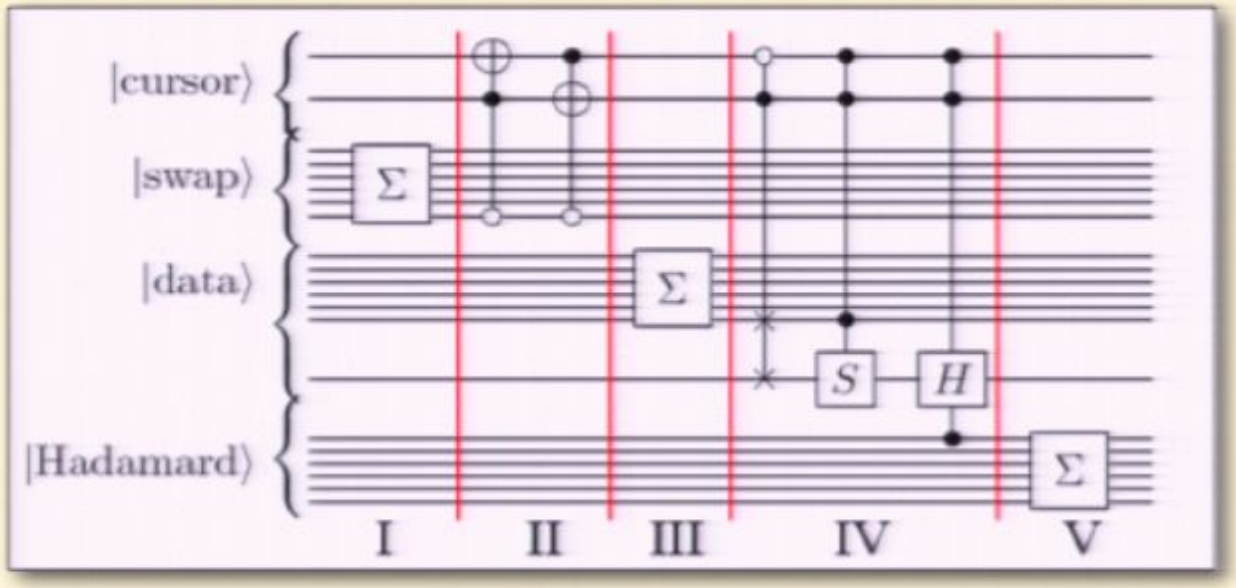
Step 2: no H

Step 3: H

SWAP program

Hadamard progra

A programmable quantum circuit



I. Cycle SWAP program qubits

II. If 0, cycle cursor program state

III. Cycle all but last data qubit

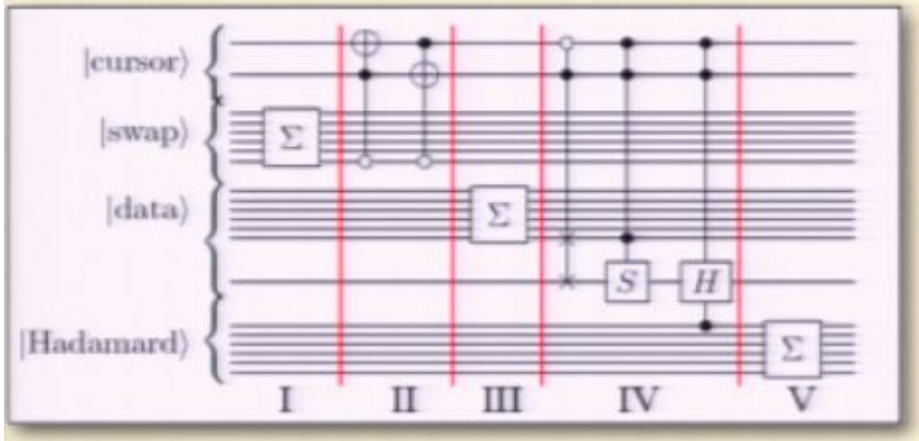
IV. If , cycle last data qubit. If , apply $\Lambda(S)$ and call Hadamard program

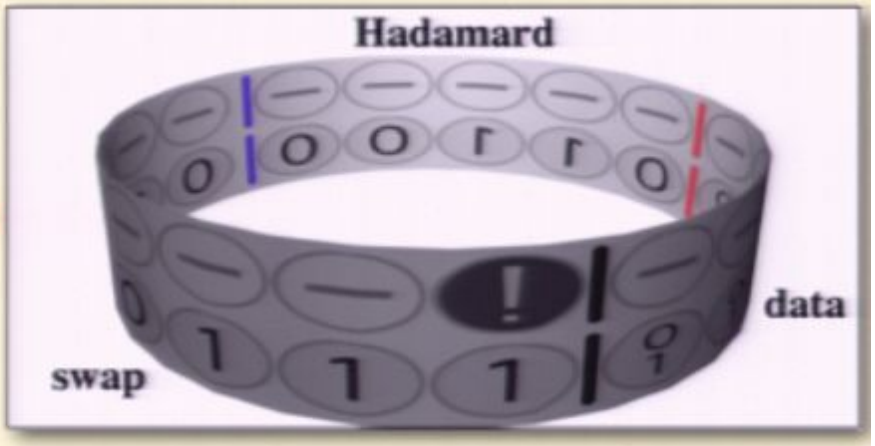
V. Cycle Hadamard program qubits

Acts as while SWAP program has 1s, then 0, then it acts as while SWAP program has 1s

Then 0, acts as , then repeats behavior

A universal spin chain quantum computer



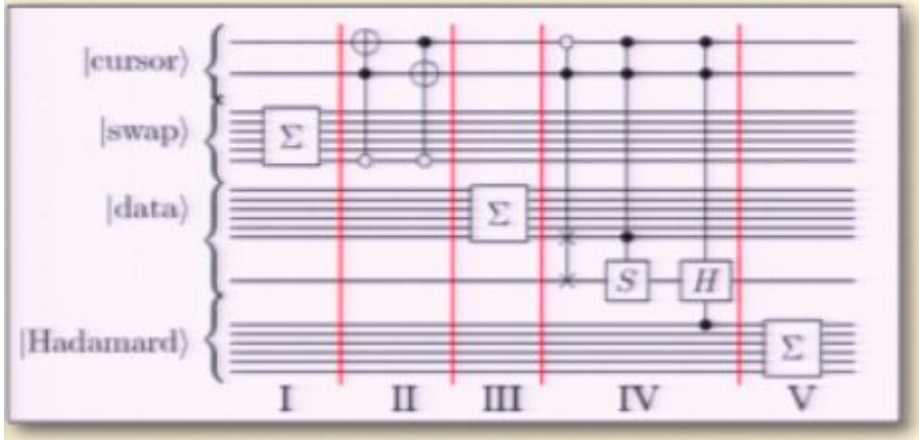


Eight states per spin: 4 for the cursor, 2 for the qubit.

$$H = \sum_{i=1}^{T} g_i + g_i^\dagger$$

2-body nearest-neighbor terms!

A universal spin chain quantum computer

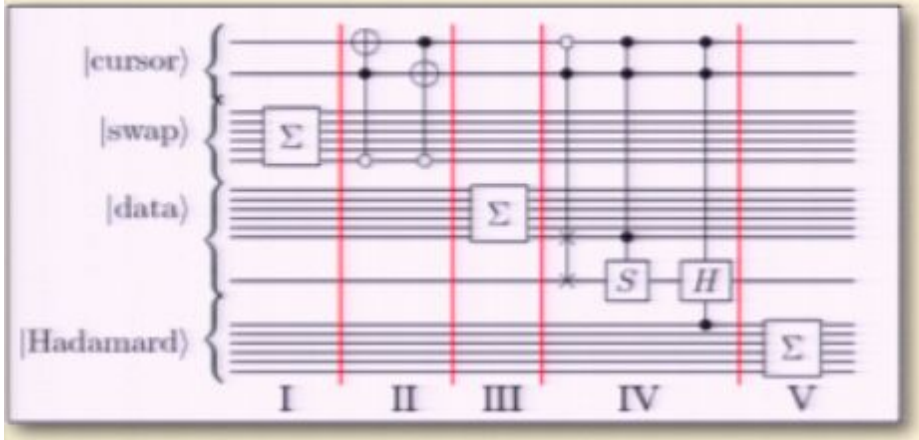


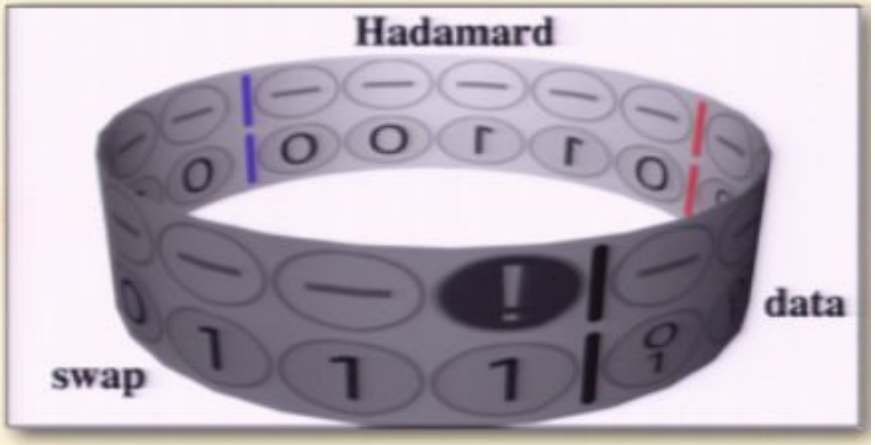
Eight states per spin: 4 for the cursor, 2 for the qubit.

$$H = \sum_{i=1}^{T} g_i + g_i^\dagger$$

2-body nearest-neighbor terms!

A universal spin chain quantum computer





Eight states per spin: 4 for the cursor, 2 for the qubit.

$$H = \sum_{i=1}^{T} g_i + g_i^\dagger$$

2-body nearest-neighbor terms!

Quantum walk analysis

$$H = \sum_{i=1}^{\overline{T}} g_i + g_i^\dagger$$

Prepare initial state $|\psi_0\rangle$ *to describe programs and data input.*

Each state $|\psi_t\rangle$ *has a unique successor state* $|\psi_{t+1}\rangle$

Size of programs = number of steps walk makes through data region; $\mathcal{O}(nT)$

"Start" and "stop" states map evolution to a quantum walk on a line of size $\overline{T} = \mathcal{O}((nT)^2)$

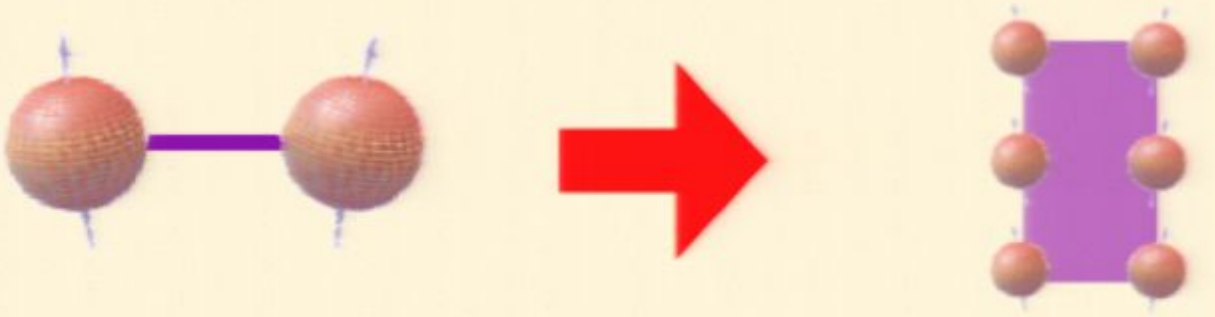
$$H_{eff} = |\psi_0\rangle\langle\psi_0| + \sum_{t=1}^{\overline{T}} (|\psi_t\rangle\langle\psi_{t-1}| + |\psi_{t-1}\rangle\langle\psi_t|) + |\psi_{\overline{T}}\rangle\langle\psi_{\overline{T}}|$$

Essentially same analysis as for Feynman's walk:

Amplitude to reach $|\psi_{\overline{T}}\rangle$ *is maximum at time* $\overline{T}/2$ *and has value* $\mathcal{O}(\overline{T}^{-1/3})$

Can engineer couplings to make simuleebaration of U exact.

Qubits? (Spin-1/2)



2-body, 8-level spin interaction

6-body, 2-level spin interaction

"Gadget" perturbation theory:

Many 2-body, NN 2-level spin interactions on a 2D grid; 6th order perturbation theory.

J. Kempe, A. Kitaev, O. Regev, SIAM J. Comput 35, 1070 (2006).
R. Oliveira, B. Terhal, arXiv:quant-ph/0504050 (2005).

Qubit spin network on the surface of a cylinder is universal!

Adiabatic quantum computing

If $\forall \delta > 0$ we have $T \geq \Omega\left(\frac{\|H(1) - H(0)\|^{1+\delta}}{\varepsilon^{\delta} \min_s\{|E_1(s) - E_2(s)|^{2+\delta}\}}\right)$,

then $|\psi_{final}\rangle$ is ε-close to the ground state of $H(1)$.

Goal:

- Want ground state of $H(0)$ to be a specified program-data state $|\psi_0\rangle$
- Want ground state of $H(1)$ to be "computation history" state

$$|\psi_{\overline{T}}\rangle = \frac{1}{\sqrt{\overline{T}+1}} \sum_{t=0}^{\overline{T}} g_t \cdots g_1 |\psi_0\rangle$$

- Want gap throughout evolution to be bounded above by a polynomial in $\overline{T}$

Can achieve this with 8-state spins using a simple variant of our Hamiltonian!

Our results in context:

- Continuous-time quantum walks with NN Hamiltonians:

  D. Janzing, PRA **75**, 012307 (2007).

  - Continuous-time quantum cellular automata with NN Hamiltonians:

    D. Nagaj, P. Wocjan, arXiv:0802.0886 (2008). NN, 10-state spins, 1D ring

    A. Kay, arXiv:0801.3228 (2008). NN, 31-state spins, 1D line

    K.G.H. Vollbrecht, J.I. Cirac, arXiv:0704.3432 (2007). NN, 30-state spins, 1D ring

    Our work: NN, 2-state spins, 2D cylinder NN, 8-state spins, 1D ring

- Adiabatic universality with NN Hamiltonians:

  D. Aharonov, D. Gottesman, S. Irani, J. Kempe, arXiv:0704.4077 (2007). NN, 9-state spins, 1D line

  Our work: NN, 8-state spins, 1D ring

- QMA-completeness of finding ground state of NN Hamiltonians:

  A. Kay, arXiv:0801.3228 (2008). Homogeneous NN, 43-state spins, 1D line

  D. Aharonov, D. Gottesman, S. Irani, J. Kempe, arXiv:0704.4077 (2007). NN, 12-state spins, 1D line

Future research directions

- Error correction and fault tolerance

Self-propelling computation + self-correcting codes?

- New program hierarchies

New architectures? Fewer states per spin?

- Algorithms directly in the model

Spatial quantum search, NAND trees, maze algorithms, etc.

- Simpler 2D construction

Without appealing to gadget perturbation theory?

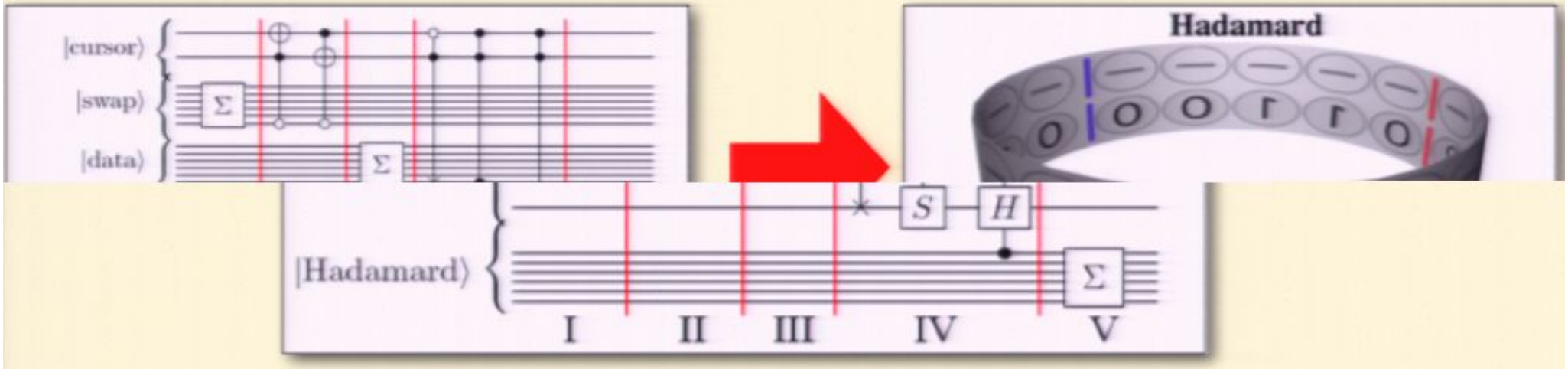
Credits

Homer Simpson.....Himself
Professor Frink.....Himself
Tweedle-dee.....Tweedle-dum
Tweedle-dum.....Tweedle-dee
Jacques Hadamard.....Himself
Bongo Player.....Richard Feynman





A universal spin chain quantum computer



I. Cycle SWAP program qubits

II. If 0, cycle cursor program state

III. Cycle all but last data qubit

IV. If , cycle last data qubit. If , apply $\Lambda(S)$ and call Hadamard program

V. Cycle Hadamard program qubits

Acts as while SWAP program has 1s, then 0, then it acts as while SWAP program has 1s

Then 0, acts as , then repeats behavior

Programmable programs

Move the next qubit B into position!

Master "cursor" program

Data qubits B A

A_1: B_1:

A_2: B_2:

SWAP program

Step 1: H

Step 2: no H

Step 3: H

Hadamard progra

A continuous-time quantum walk computer

Foundations of Physics, Vol. 16, No. 6, 1986

Quantum Mechanical Computers[1]

Richard P. Feynman[2]

Received March 15, 1985

The physical limitations, due to quantum mechanics, on the functioning of computers are analyzed.

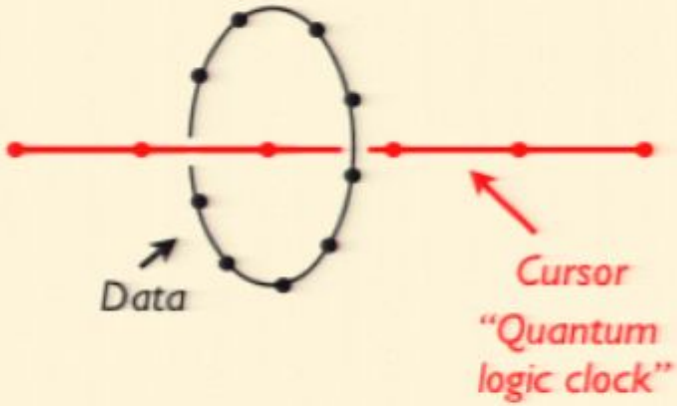
[1] *Editor's note*: This article, which is based on the author's plenary talk presented at the CLEO/IQEC Meeting in 1984, originally appeared in the February 1985 issue of *Optics News*. It is here reprinted with kind permission of Professor Feynman and *Optics News*.

[2] Department of Physics, California Institute of Technology, Pasadena, California 91125.

Standard quantum circuit: $U_T \cdots U_1$

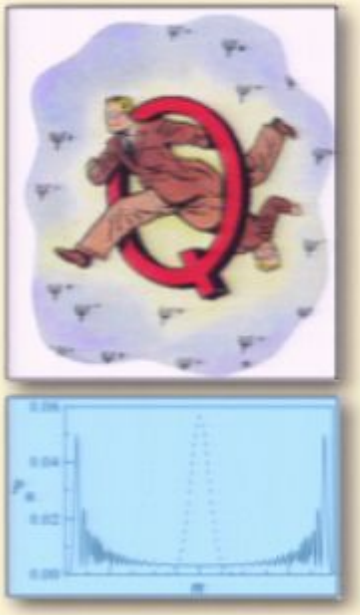
Feynman Hamiltonian:

$$H_F = \sum_{t=1}^{T} U_t \sigma_t^+ \sigma_{t-1}^- + U_t^\dagger \sigma_{t-1}^- \sigma_t^+$$



Quantum walks

Goal: Design a programmably universal time-independent nearest-neighbor spin-chain Hamiltonian.



Evolution by a time-independent Hamiltonian:

Continuous-time quantum walk on a graph.

Continuous-time random walk:

$$\frac{d\vec{p}}{dt} = -K\vec{p}$$

Probability conservation:

$$\frac{d\,\|\vec{p}\,\|_1}{dt} = 0$$

$$K_{ij} \in \mathbb{R} \qquad \sum_j K_{ij} = 0$$

Continuous-time quantum walk:

$$\frac{d|\psi\rangle}{dt} = -iH|\psi\rangle$$

Amplitude conservation:

$$\frac{d\,\||\psi\rangle\|_2}{dt} = 0$$

$$H_{ij} \in \mathbb{C} \qquad H_{ij} = H_{ji}^*$$