

**Formal
Verification**



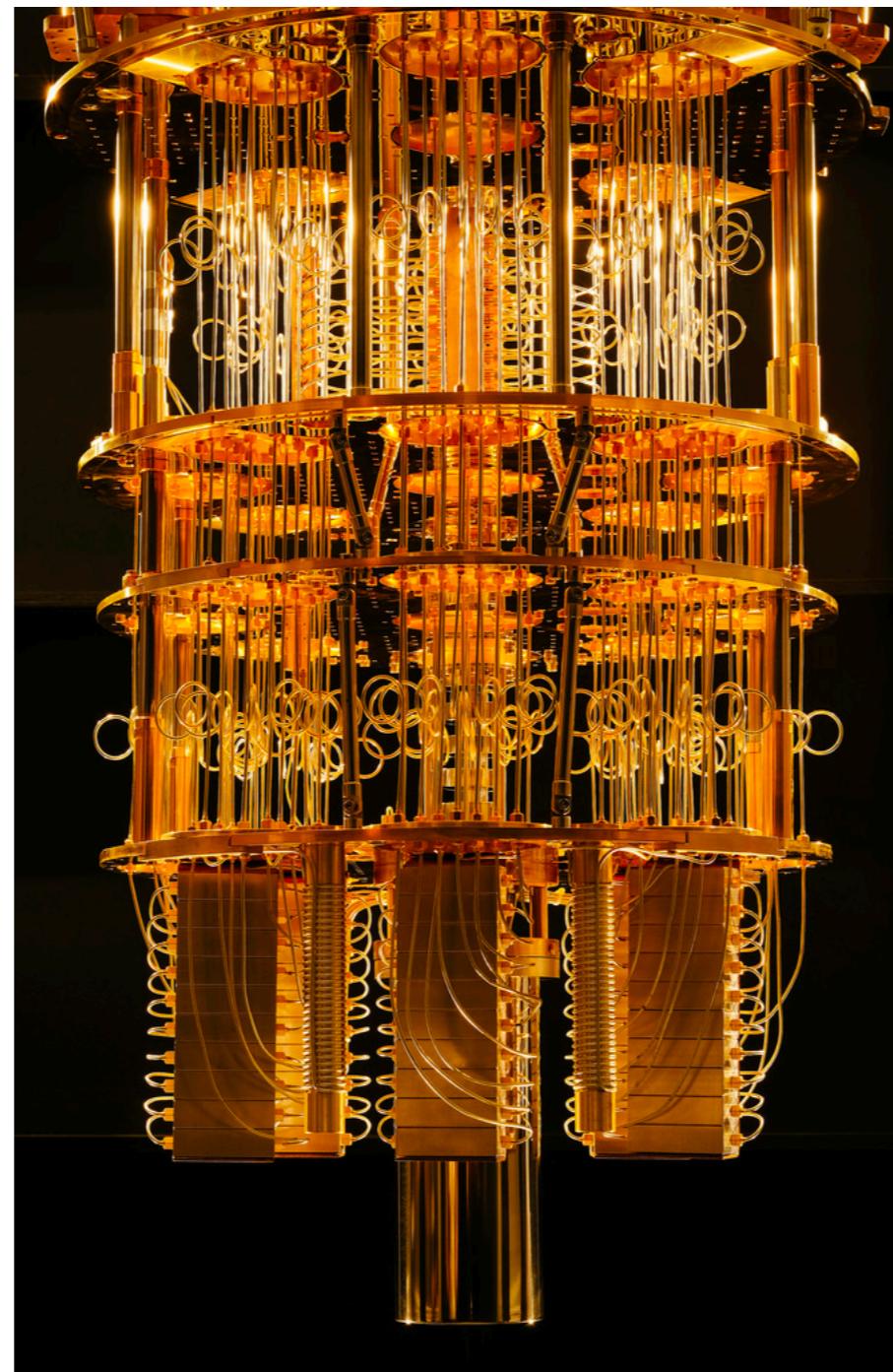
**Quantum
Uncertainty**



SNAPL 2019

Robert Rand, Kesha Heitala and *Michael Hicks*

Quantum Computing



Quantum Computing

Quantum Computing

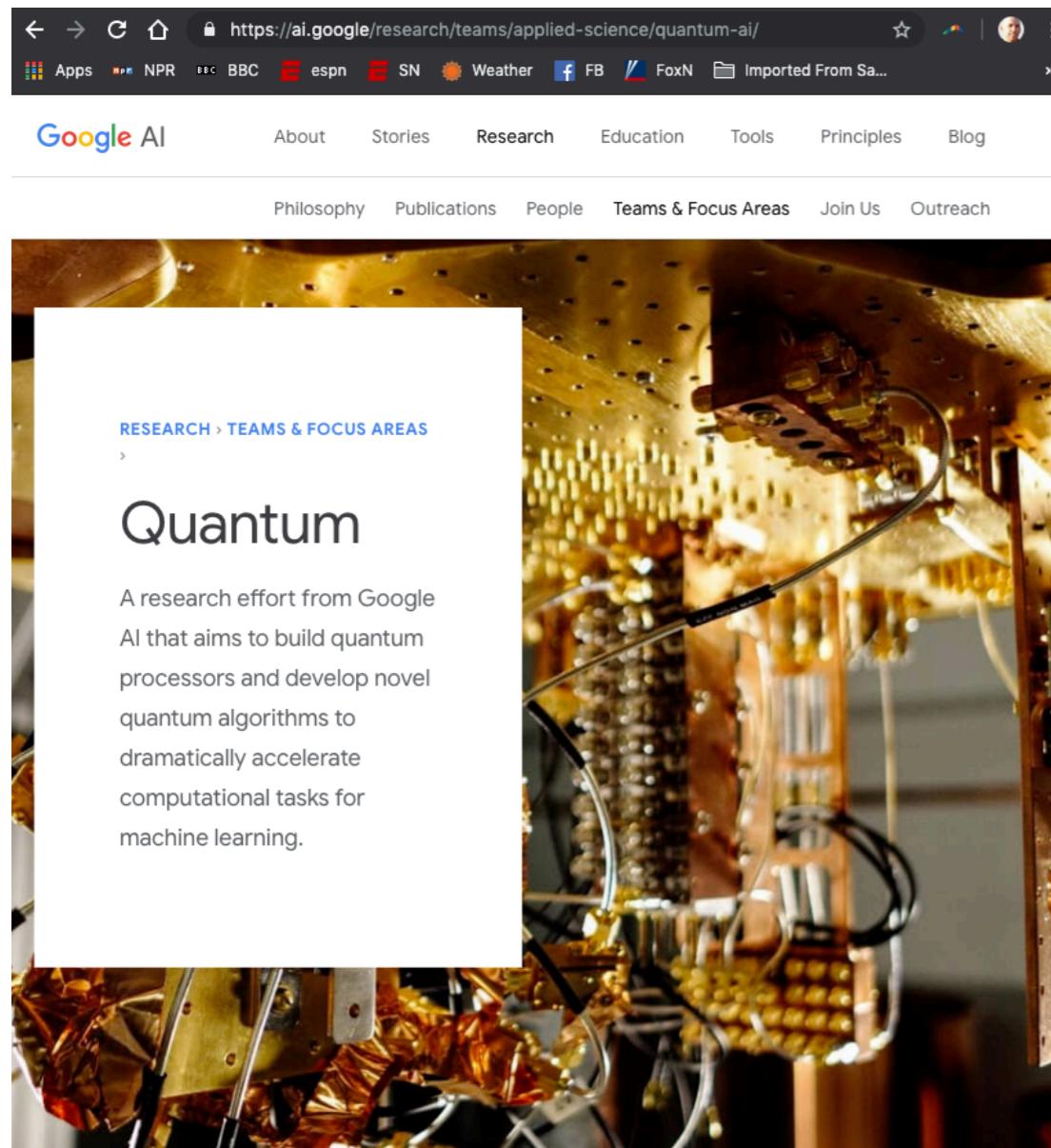
The screenshot shows a web browser window with the URL <https://ai.google/research/teams/applied-science/quantum-ai/>. The page is titled "Quantum" and describes it as a research effort from Google AI focused on building quantum processors and developing novel quantum algorithms for machine learning. To the right of the text is a close-up photograph of a complex quantum computing hardware setup, featuring numerous gold-colored metal components, wires, and optical elements.

RESEARCH > TEAMS & FOCUS AREAS

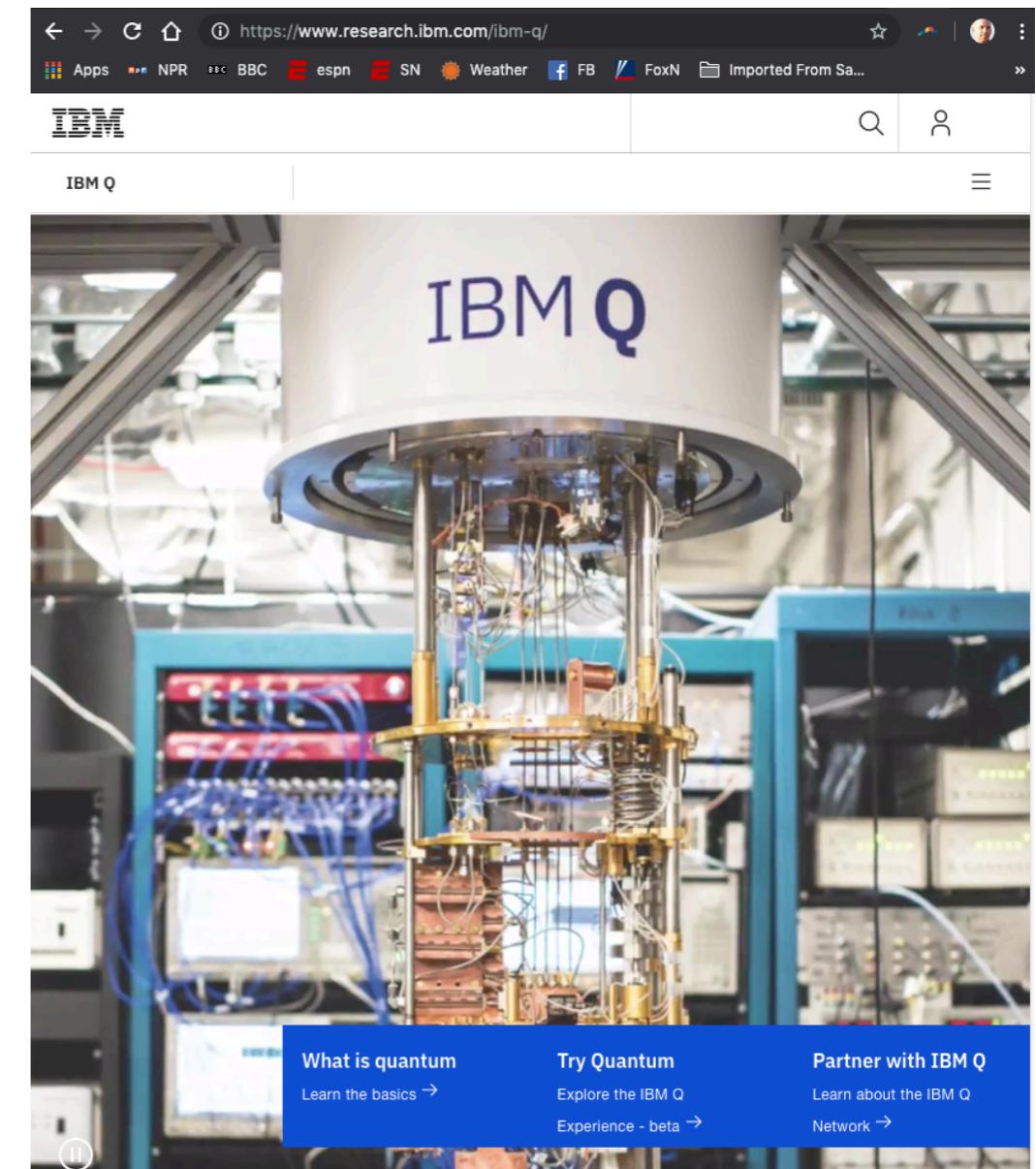
Quantum

A research effort from Google AI that aims to build quantum processors and develop novel quantum algorithms to dramatically accelerate computational tasks for machine learning.

Quantum Computing



The screenshot shows a web browser window for the URL <https://ai.google/research/teams/applied-science/quantum-ai/>. The page header includes the Google AI logo and navigation links for About, Stories, Research, Education, Tools, Principles, Blog, Philosophy, Publications, People, Teams & Focus Areas, Join Us, and Outreach. A large image of a quantum computing hardware setup is displayed, showing a complex array of gold-colored components and wires. A white sidebar on the left contains the text "RESEARCH > TEAMS & FOCUS AREAS" and "Quantum". Below this, a detailed description reads: "A research effort from Google AI that aims to build quantum processors and develop novel quantum algorithms to dramatically accelerate computational tasks for machine learning."



The screenshot shows a web browser window for the URL <https://www.research.ibm.com/ibm-q/>. The page header includes the IBM logo and navigation links for Apps, NPR, BBC, ESPN, SN, Weather, Facebook, FoxN, and Imported From Sa... The main content features a large image of a quantum computing system with the "IBM Q" logo prominently displayed. At the bottom of the page, there is a blue footer bar with three items: "What is quantum" (with a link to "Learn the basics →"), "Try Quantum" (with a link to "Explore the IBM Q Experience - beta →"), and "Partner with IBM Q" (with links to "Learn about the IBM Q Network →").

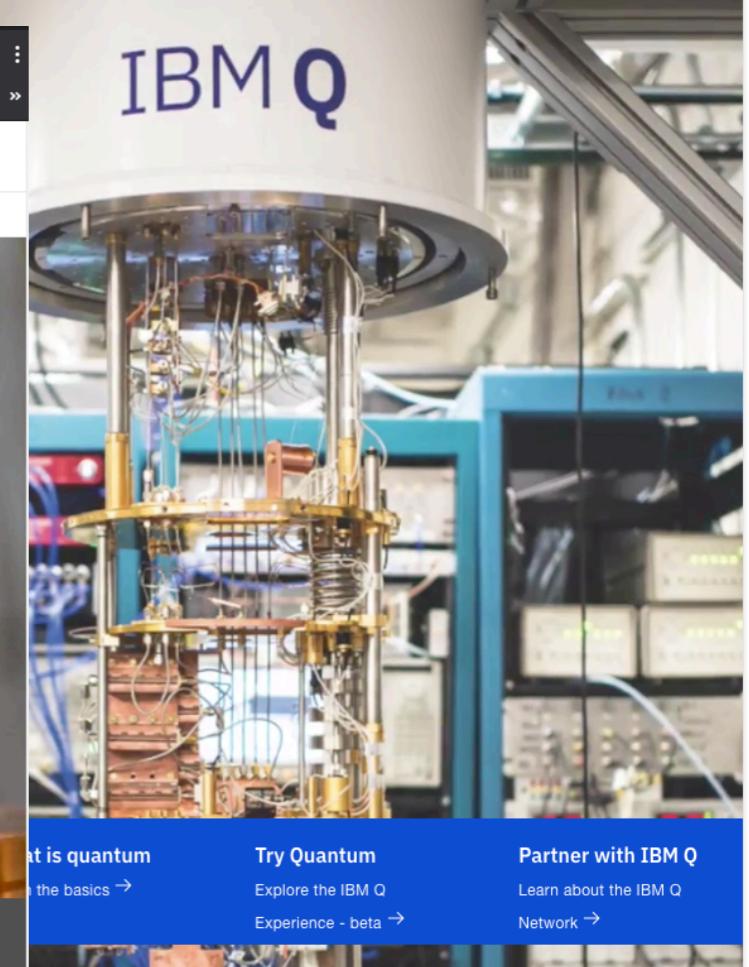
Quantum Computing

The screenshot shows the Google AI website at <https://ai.google/research/teams/applied-science/quantum-ai/>. The page features a navigation bar with links to About, Stories, Research, Education, Tools, Principles, Blog, Philosophy, Publications, People, Teams & Focus Areas, Join Us, and Outreach. A large image of a quantum computing hardware setup is visible on the left.

The screenshot shows the IBM Q website at <https://www.research.ibm.com/ibm-q/>. The page features a navigation bar with links to Apps, NPR, BBC, ESPN, SN, Weather, Facebook, FoxN, and Imported From Sa... The main content area displays a large image of a quantum computing hardware setup with the "IBM Q" logo prominently displayed.



The screenshot shows the Microsoft Quantum website at <https://www.microsoft.com/en-us/quantum/>. The page features a navigation bar with links to Apps, NPR, BBC, ESPN, SN, Weather, Facebook, FoxN, and Imported From Sa... The main content area displays a large image of a quantum computing hardware setup with the text "Empowering the quantum revolution" and "Your path to powerful, scalable quantum computing starts here." A "Watch now" button is present. At the bottom, there is a call-to-action: "Join us at the leading edge of opportunity".



Quantum Computing

The screenshot shows a web browser window with the URL <https://www.energy.gov/articles/department-energy-announces-218-million>. The page is from the ENERGY.GOV website, featuring a green header with the Department of Energy logo and the title "Department of Energy Announces \$218 Million for Quantum Information Science". The date "SEPTEMBER 24, 2018" is below the title. Social sharing icons for email, Facebook, Twitter, LinkedIn, and Pinterest are present. A sidebar on the left discusses Google AI research teams and quantum computing, mentioning a \$218 million investment. Another sidebar on the right promotes IBM Q and partner opportunities. The main content area includes a photograph of quantum computing equipment.

RESEARCH > TEAMS & FOCUS AREAS

Quantum

A research effort from Google AI that aims to build quantum processors and develop novel quantum algorithms to dramatically accelerate computational tasks for machine learning.

Home » Department of Energy Announces \$218 Million for Quantum Information Science

Field Will Shape Future of Information Processing

WASHINGTON, D.C. – Today, the U.S. Department of Energy (DOE) announced \$218 million in funding for 85 research awards in the important emerging field of Quantum Information Science (QIS). The awards were made in conjunction with the White House Summit on Advancing American Leadership in QIS, highlighting

software, from development through deployment, Microsoft brings the only scalable quantum system to the broadest set of customers.

Try Quantum
Explore the IBM Q Experience - beta →

Partner with IBM Q
Learn about the IBM Q Network →

Why?

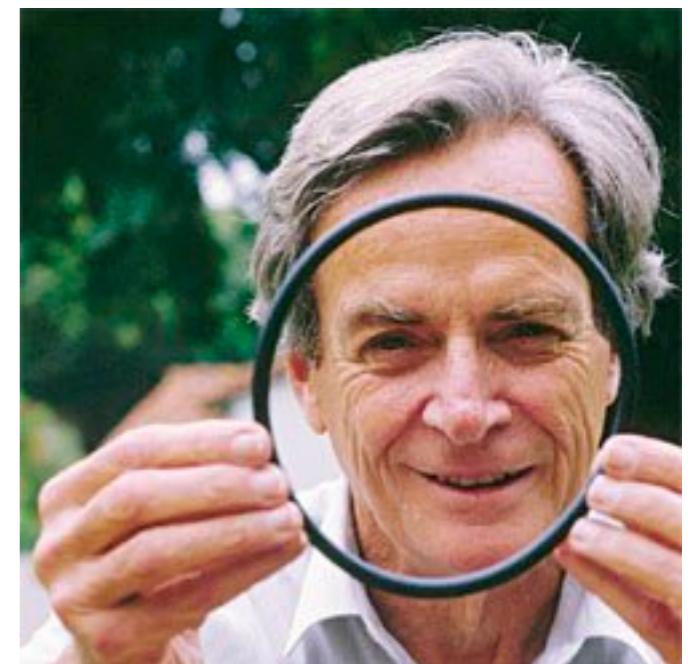
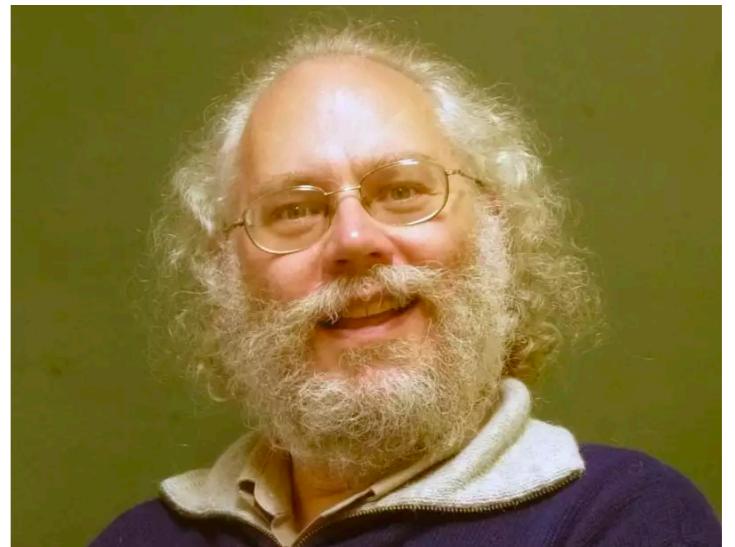
Why?

- Factoring large numbers



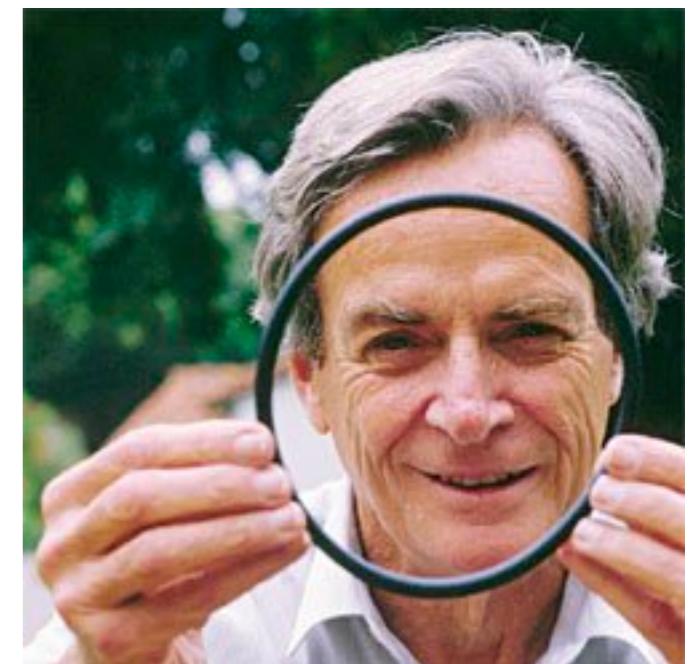
Why?

- Factoring large numbers
- Simulating small physical systems



Why?

- Factoring large numbers
- Simulating small physical systems
- Uncrackable crypto systems
- Solving linear equations



Challenges

In general

- Answer may be unknown
- Simulation is intractable
- Breakpoints break things (opening the box kills the cat)

In the near term

- Execution is expensive, and (highly) error prone
- Computing resources (e.g., qubits) are scarce

PL to the Rescue?

PL to the Rescue?

- Compilers and optimizations, language design, formal methods, approximate computing, probabilistic programming, ...
 - Sound familiar?

PL to the Rescue?

- Compilers and optimizations, language design, formal methods, approximate computing, probabilistic programming, ...
 - Sound familiar?
- Goal of this talk: Get you interested to work on QC with your PL hat on !

Qubits

Superposition: Qubits can be in multiple states (0 or 1) at once

Qubits

$$\begin{pmatrix} \alpha \\ \beta \end{pmatrix}$$

Superposition: Qubits can be in multiple states (0 or 1) at once

Qubits

$$\begin{pmatrix} \alpha \\ \beta \end{pmatrix}$$

$$|\alpha|^2 + |\beta|^2 = 1$$

Superposition: Qubits can be in multiple states (0 or 1) at once

Qubits

$$\begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

$$\begin{pmatrix} \alpha \\ \beta \end{pmatrix}$$

$|0\rangle$

$$|\alpha|^2 + |\beta|^2 = 1$$

Superposition: Qubits can be in multiple states (0 or 1) at once

Qubits

$$\begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad \begin{pmatrix} \alpha \\ \beta \end{pmatrix} \quad \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$
$$|0\rangle \qquad \qquad \qquad |1\rangle$$
$$|\alpha|^2 + |\beta|^2 = 1$$

Superposition: Qubits can be in multiple states (0 or 1) at once

Measurement

Measurement: Looking at a qubit probabilistically turns it into a bit.

Measurement

$$\begin{pmatrix} \alpha \\ \beta \end{pmatrix}$$

Measurement: Looking at a qubit probabilistically turns it into a bit.

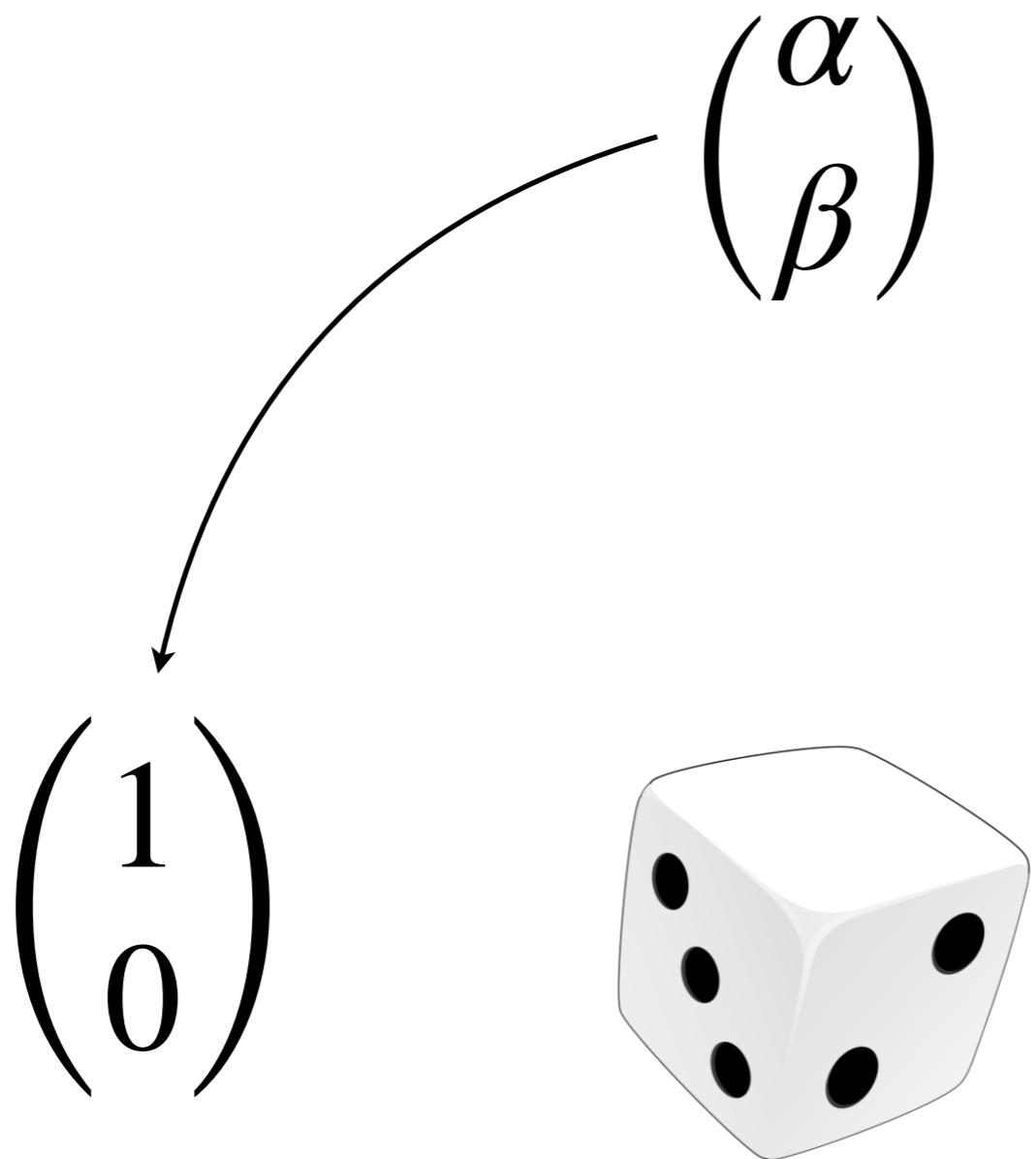
Measurement

$$\begin{pmatrix} \alpha \\ \beta \end{pmatrix}$$



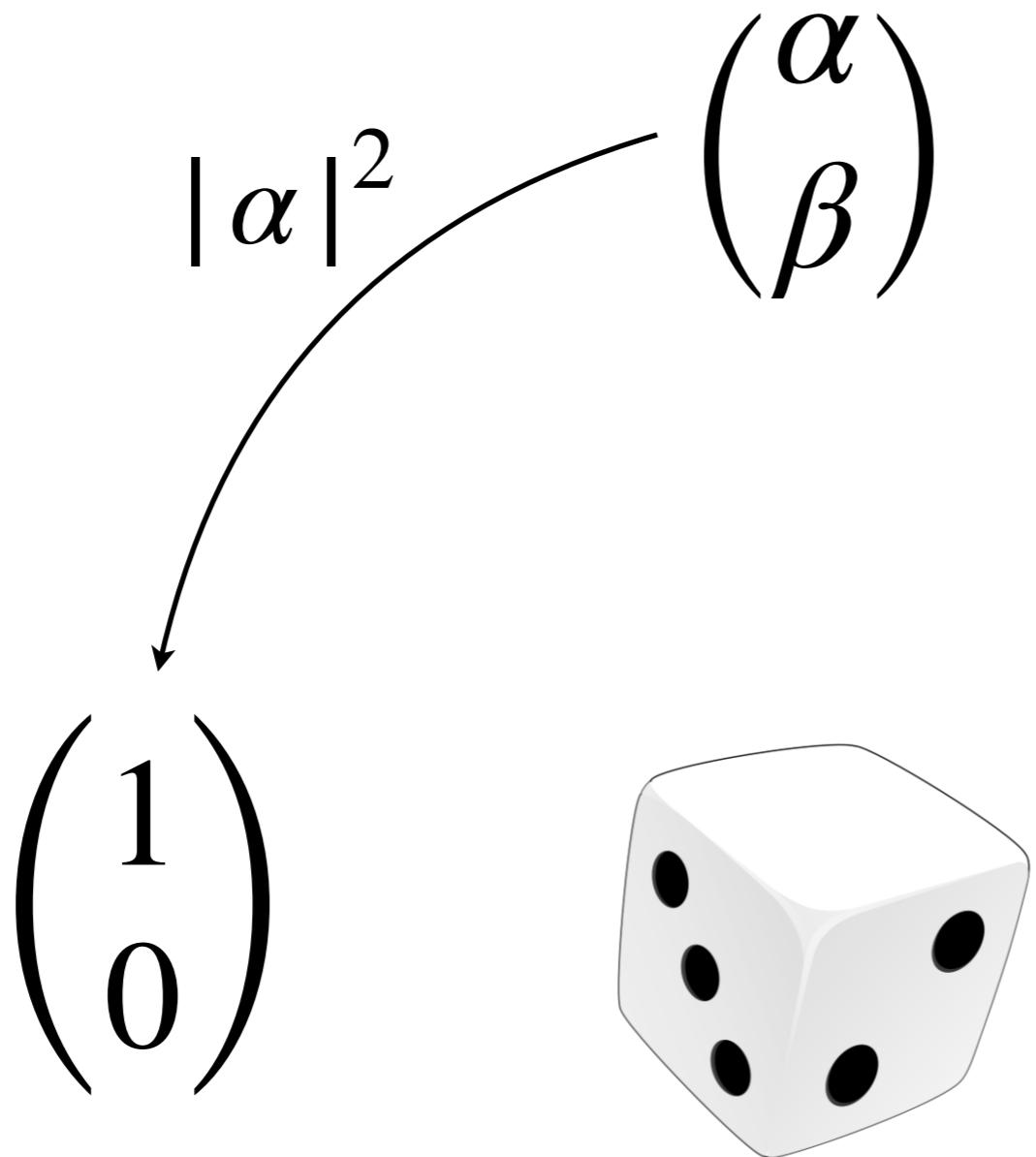
Measurement: Looking at a qubit probabilistically turns it into a bit.

Measurement



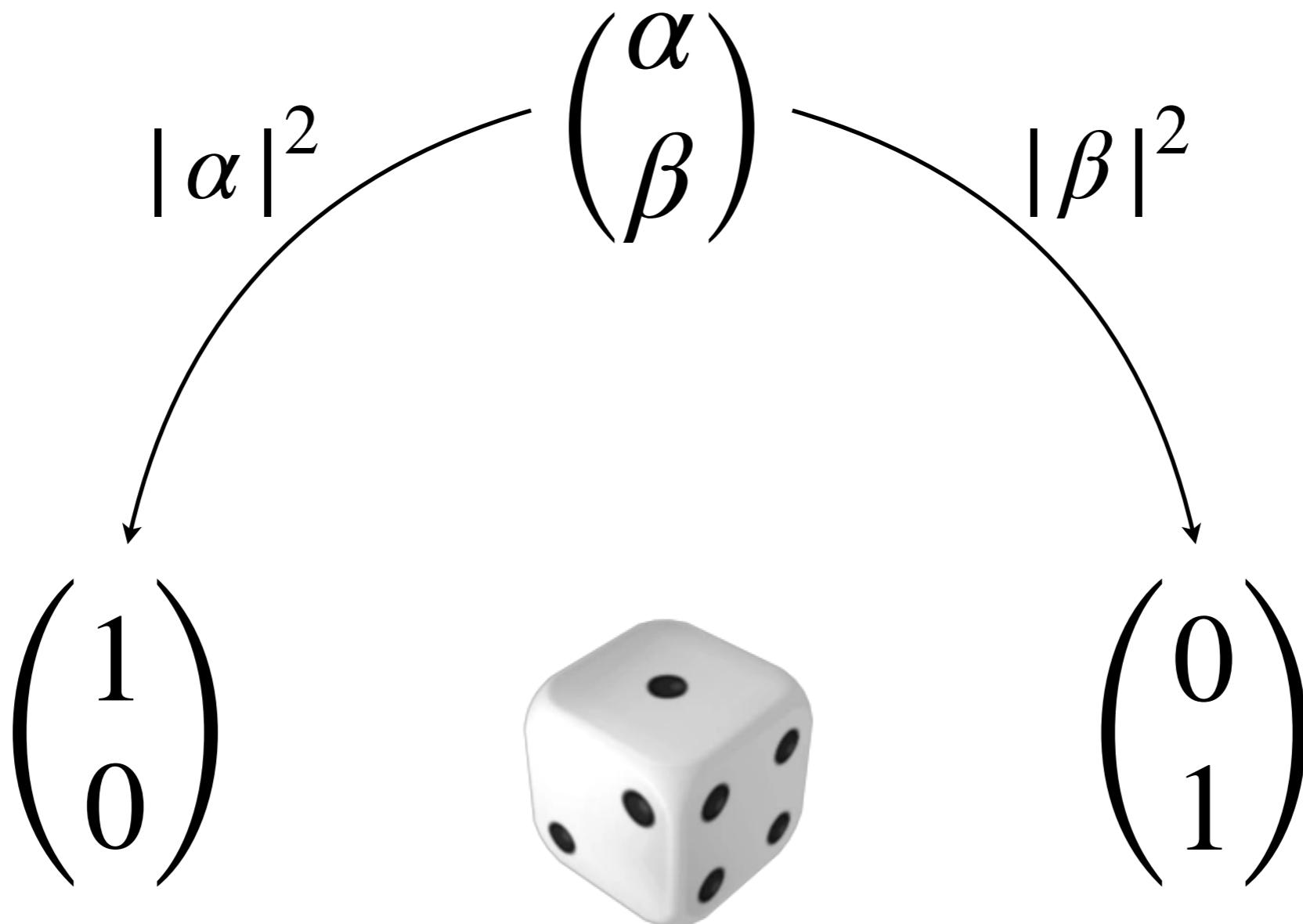
Measurement: Looking at a qubit probabilistically turns it into a bit.

Measurement



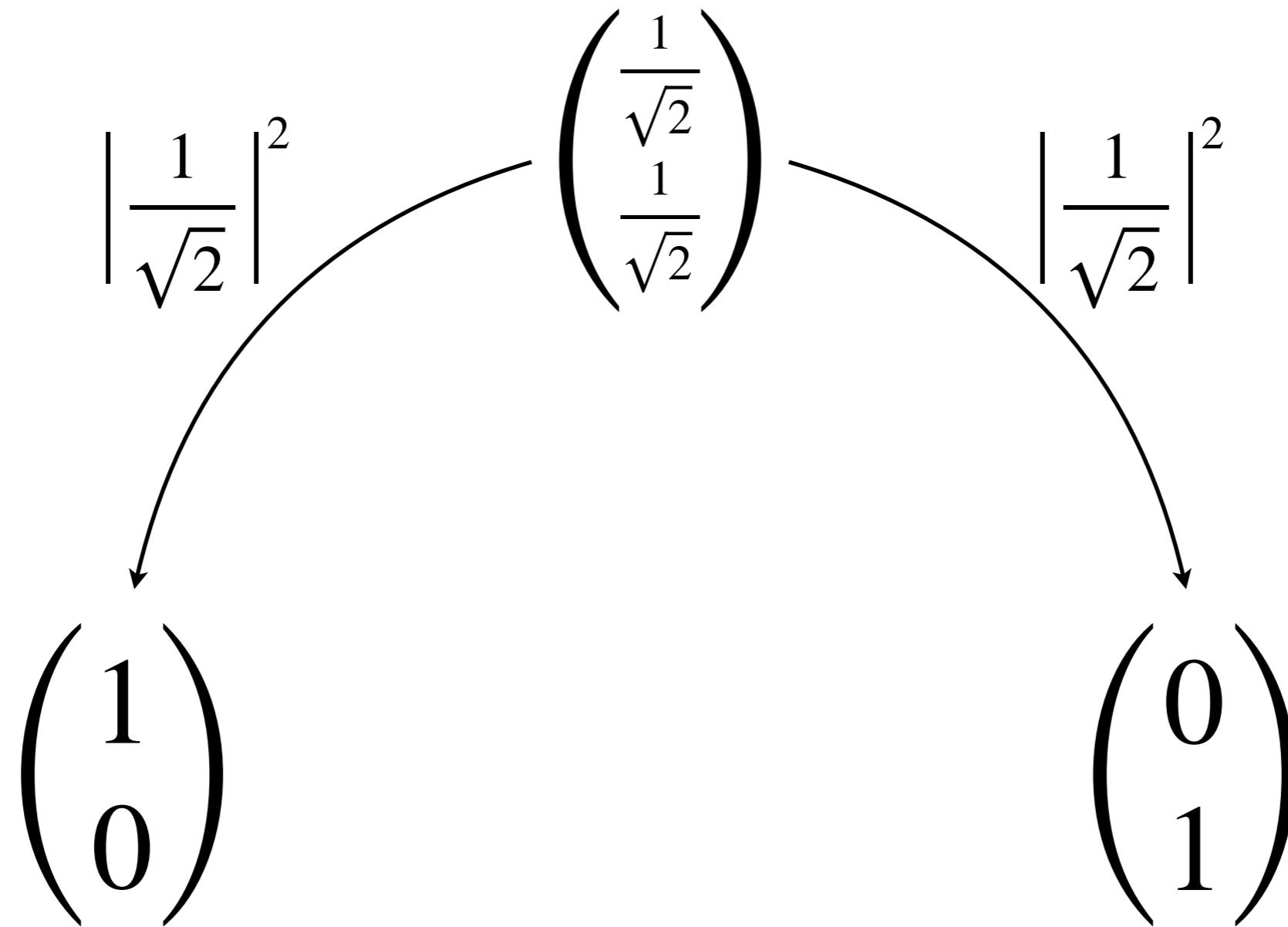
Measurement: Looking at a qubit probabilistically turns it into a bit.

Measurement

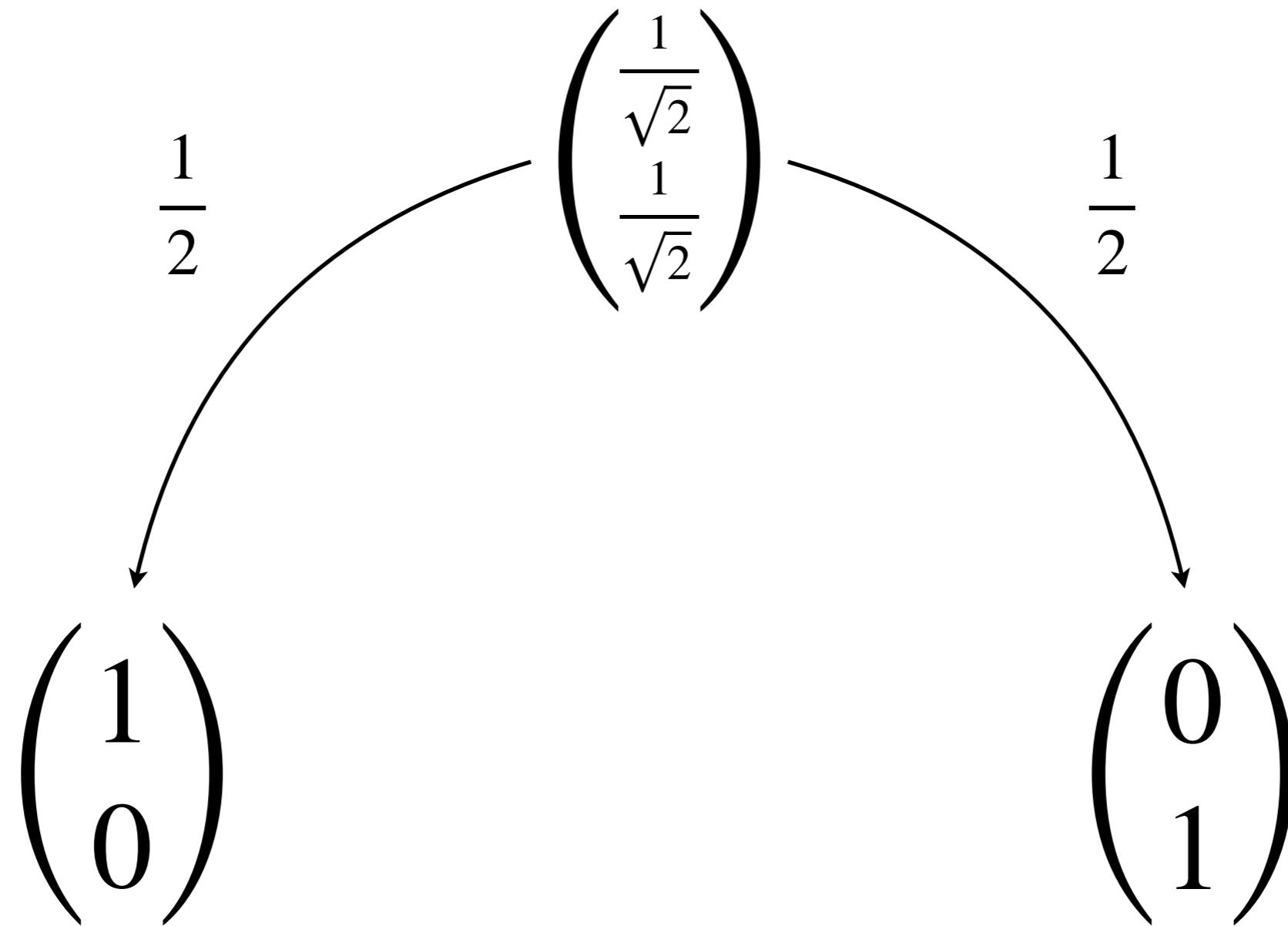


Measurement: Looking at a qubit probabilistically turns it into a bit.

Measurement



Measurement



Unitaries

$$\begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

Unitaries are operators that transform (“evolve”) quantum states

Unitaries

$$\frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

Unitaries are operators that transform (“evolve”) quantum states

Unitaries

$$\frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix}$$

Unitaries are operators that transform (“evolve”) quantum states

Unitaries

$$\frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix}$$

$$\frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix}$$

Unitaries are operators that transform (“evolve”) quantum states

Unitaries

$$\frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix}$$

$$\frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

Unitaries are operators that transform (“evolve”) quantum states

Unitaries

$$H|0\rangle = |+\rangle$$

$$\frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

This is the *Hadamard* unitary, labeled H

Unitaries

$$H |0\rangle = |+\rangle$$

$$H |+\rangle = |0\rangle$$

This is the *Hadamard* unitary, labeled H

Multiple Qubits

$$\begin{pmatrix} \frac{1}{\sqrt{2}} \\ 0 \\ \frac{1}{\sqrt{2}} \end{pmatrix} \otimes \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

Multi-qubit states can be formed via the tensor product

Multiple Qubits

$$\begin{pmatrix} \frac{1}{\sqrt{2}} \\ 0 \\ \frac{1}{\sqrt{2}} \end{pmatrix} \otimes \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ \frac{1}{\sqrt{2}} \\ 0 \\ \frac{1}{\sqrt{2}} \end{pmatrix}$$

Multi-qubit states can be formed via the tensor product

Multiple Qubits

$$\begin{pmatrix} \frac{1}{\sqrt{2}} \\ 0 \\ \frac{1}{\sqrt{2}} \end{pmatrix} \otimes \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ \frac{1}{\sqrt{2}} \\ 0 \\ \frac{1}{\sqrt{2}} \end{pmatrix}$$

$$|+\rangle \otimes |0\rangle = |+\rangle |0\rangle$$

Multi-qubit states can be formed via the tensor product

Multiqubit Unitaries

$$\begin{pmatrix} 0 \\ \frac{1}{\sqrt{2}} \\ 0 \\ \frac{1}{\sqrt{2}} \end{pmatrix}$$

Multiquubit Unitaries

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ \frac{1}{\sqrt{2}} \\ 0 \\ \frac{1}{\sqrt{2}} \end{pmatrix}$$

Multiqubit Unitaries

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ \frac{1}{\sqrt{2}} \\ 0 \\ \frac{1}{\sqrt{2}} \end{pmatrix} = \begin{pmatrix} 0 \\ \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \\ 0 \end{pmatrix}$$

Multiqubit Unitaries

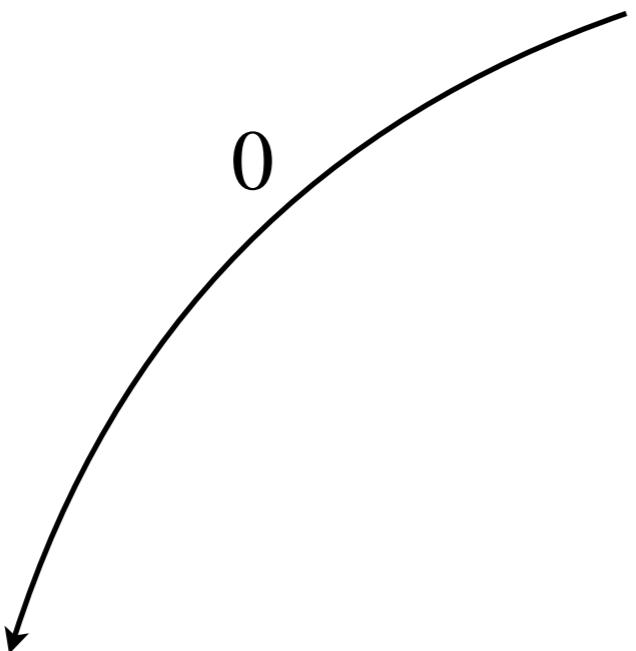
$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ \frac{1}{\sqrt{2}} \\ 0 \\ \frac{1}{\sqrt{2}} \end{pmatrix} = \begin{pmatrix} 0 \\ \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \\ 0 \end{pmatrix}$$

$$\text{CNOT} |+\rangle |0\rangle = \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle)$$

Measurement 2.0

$$\begin{pmatrix} 0 \\ \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \\ 0 \end{pmatrix}$$

Measurement 2.0


$$\begin{pmatrix} 0 \\ \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \\ 0 \end{pmatrix}$$

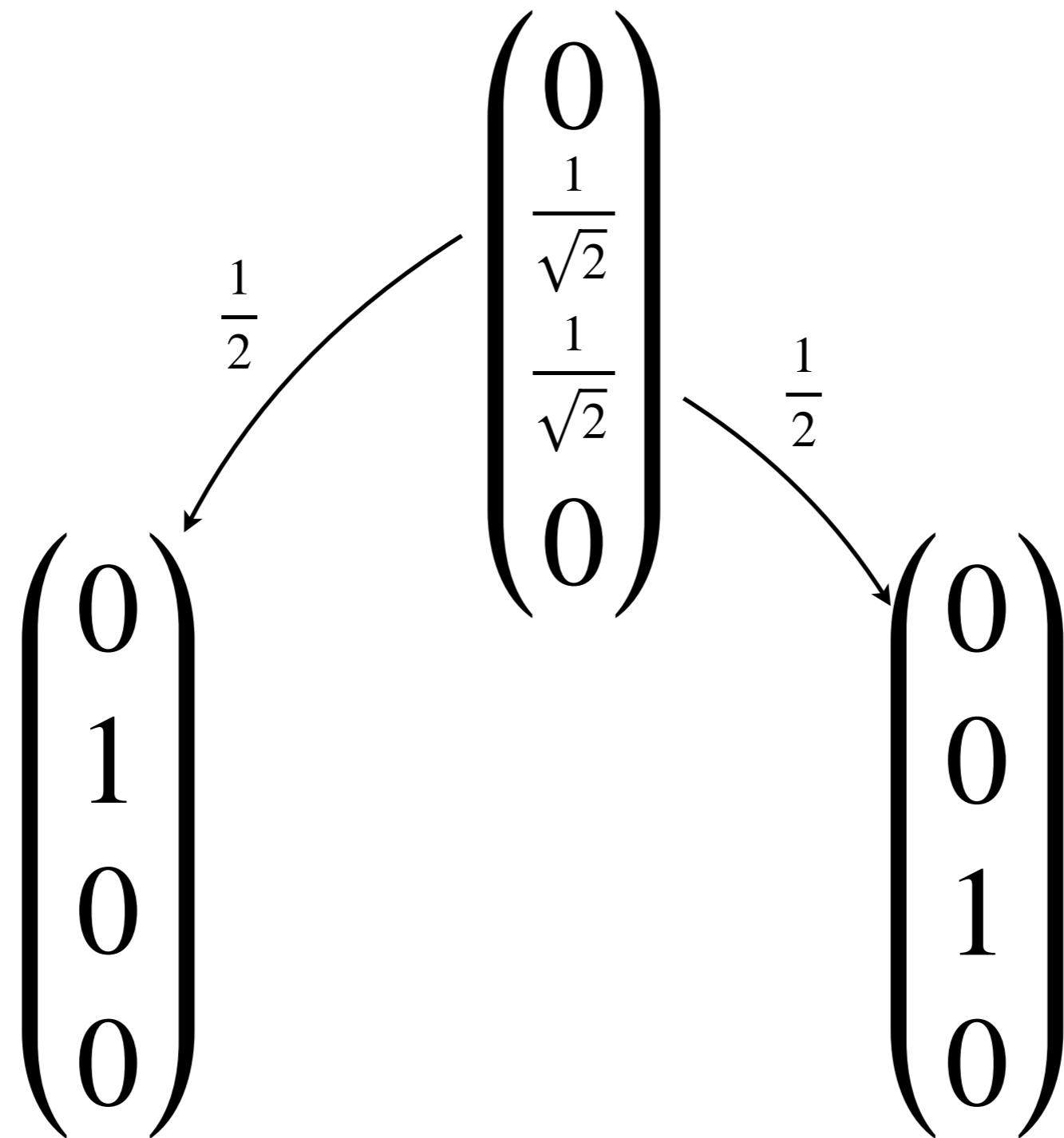
Measurement 2.0

$$\begin{pmatrix} 0 \\ \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \\ 0 \end{pmatrix}$$

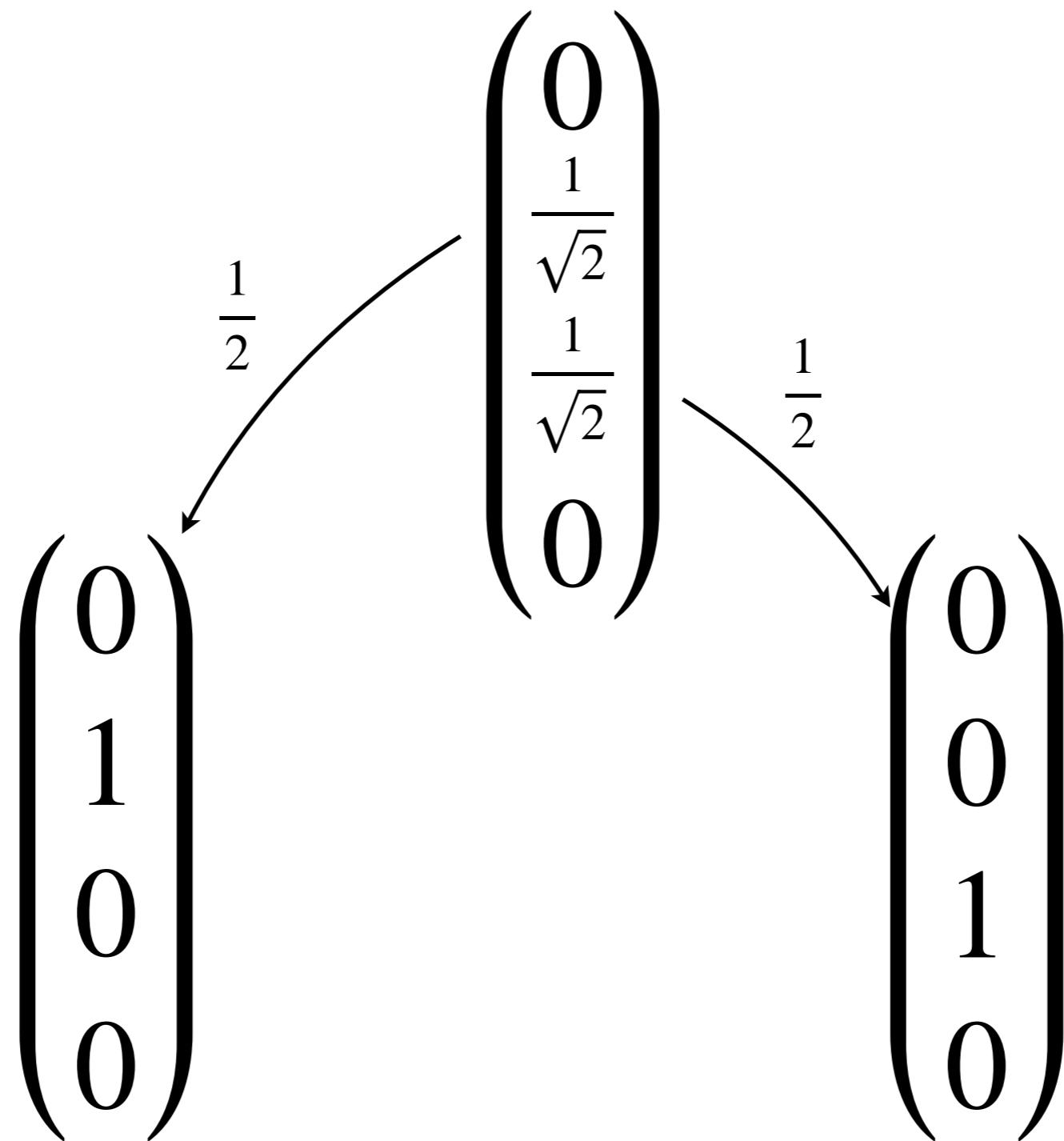
Measurement 2.0

$$\begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} \xrightarrow{\frac{1}{2}} \begin{pmatrix} 0 \\ \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \\ 0 \end{pmatrix}$$

Measurement 2.0



Measurement 2.0



Measurement 2.0

$$\begin{pmatrix} 1 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 0 \\ 1 \end{pmatrix} \xrightarrow{\frac{1}{2}} \begin{pmatrix} 0 \\ \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \\ 0 \end{pmatrix} \xrightarrow{\frac{1}{2}} \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}$$

Measurement 2.0

$$\begin{pmatrix} 1 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 0 \\ 1 \end{pmatrix} \xrightarrow{\frac{1}{2}} \begin{pmatrix} 0 \\ \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \\ 0 \end{pmatrix} \xrightarrow{\frac{1}{2}} \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

Measurement 2.0

$$\begin{pmatrix} 1 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 0 \\ 1 \end{pmatrix} \quad \left| 01 \right\rangle$$
$$\begin{pmatrix} 0 \\ 1 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad \left| 10 \right\rangle$$

A diagram illustrating a quantum measurement process. At the top center is a vertical vector labeled $\begin{pmatrix} 0 \\ \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \\ 0 \end{pmatrix}$. Two curved arrows point downwards from this vector to two separate terms below. The left arrow points to the term $\begin{pmatrix} 1 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 0 \\ 1 \end{pmatrix}$, which is associated with the state $\left| 01 \right\rangle$ at the bottom. The right arrow points to the term $\begin{pmatrix} 0 \\ 1 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 0 \end{pmatrix}$, which is associated with the state $\left| 10 \right\rangle$ at the bottom.

Entanglement

$$\begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}$$

Entanglement

$$\begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

Entanglement

$$\begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \\ 0 \end{pmatrix}$$

Entanglement

$$\begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}$$

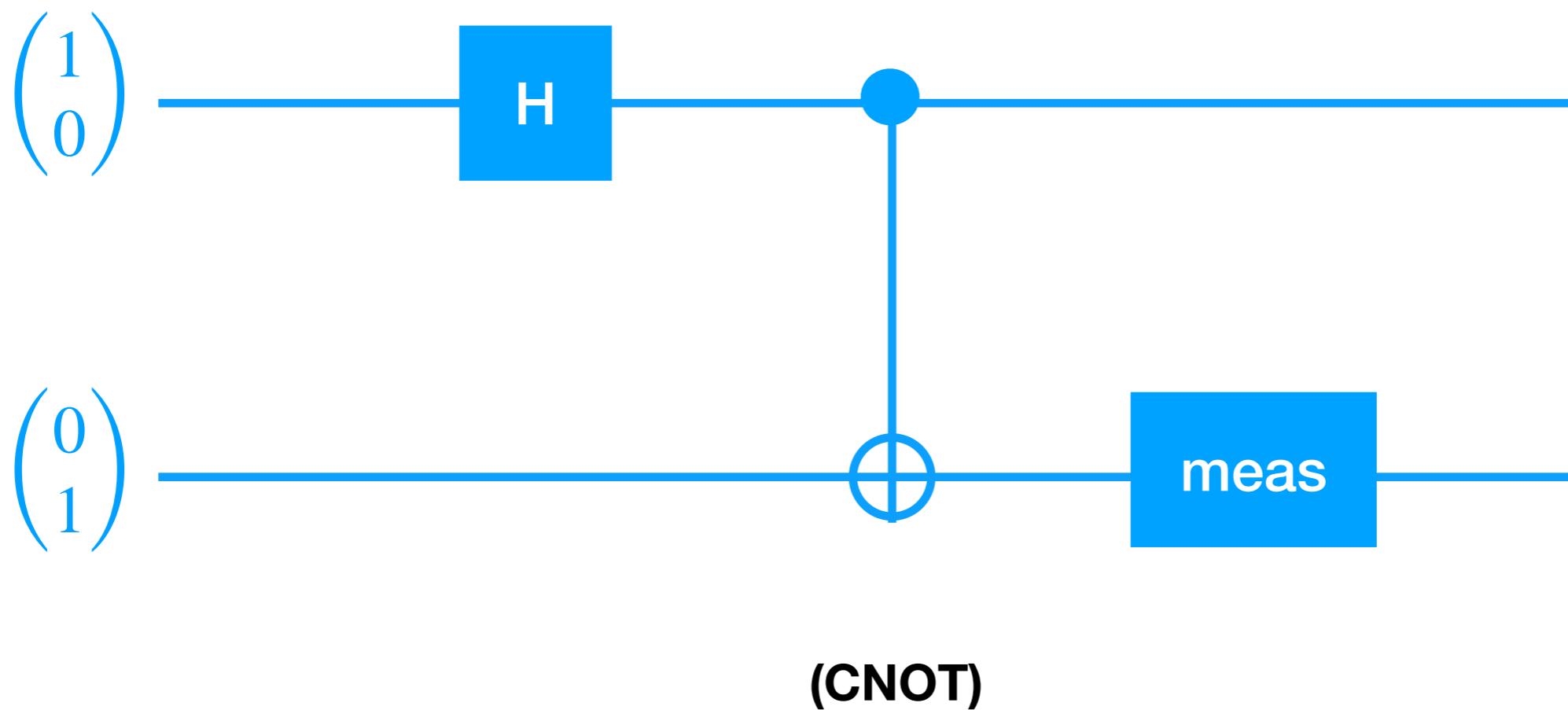
$$\begin{pmatrix} 1 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

$$\begin{pmatrix} 0 \\ \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \\ 0 \end{pmatrix}$$

$$? \otimes ?$$

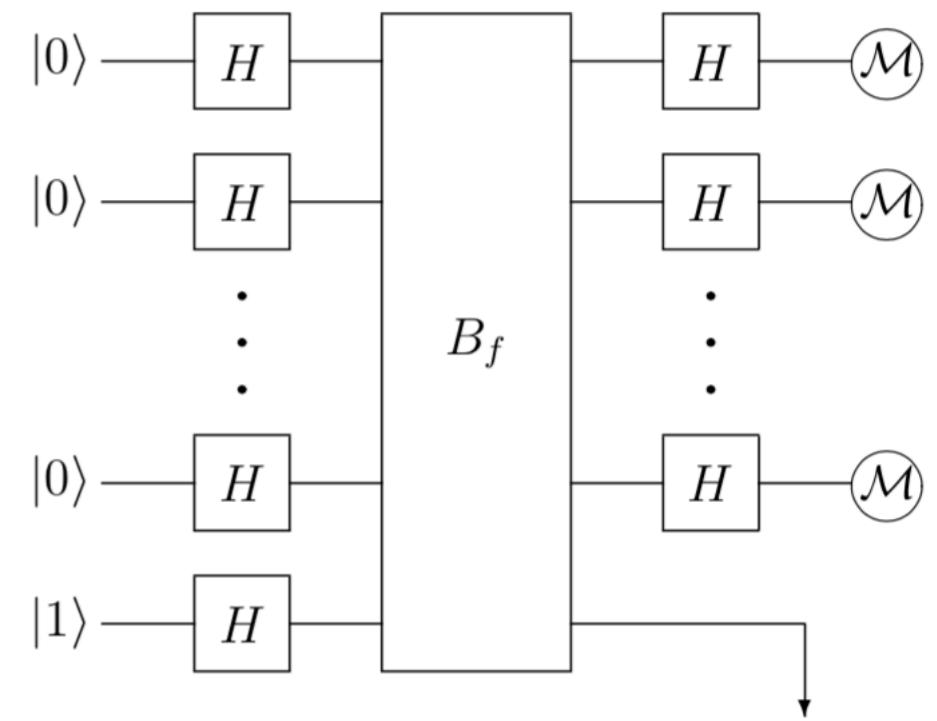
Entangled qubits are not probabilistically independent – they cannot be decomposed. Connection at a distance!

Circuits



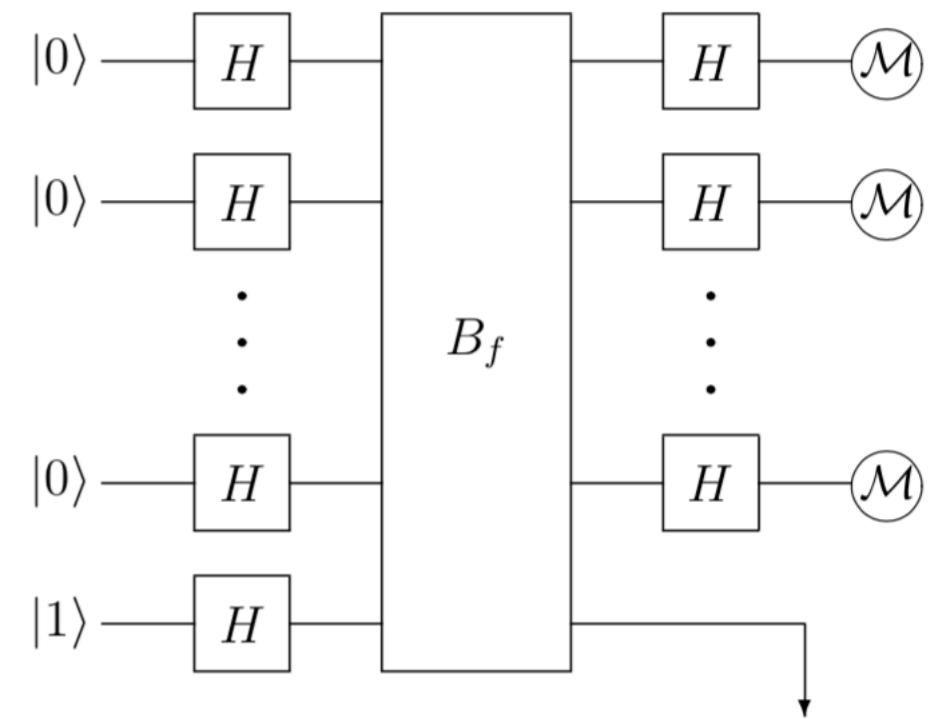
Algorithms

- Common theme of many quantum algorithms
 - Compute over the state space in superposition, essentially doing work “in parallel”
 - measure to produce the final result (and cancel out noise)



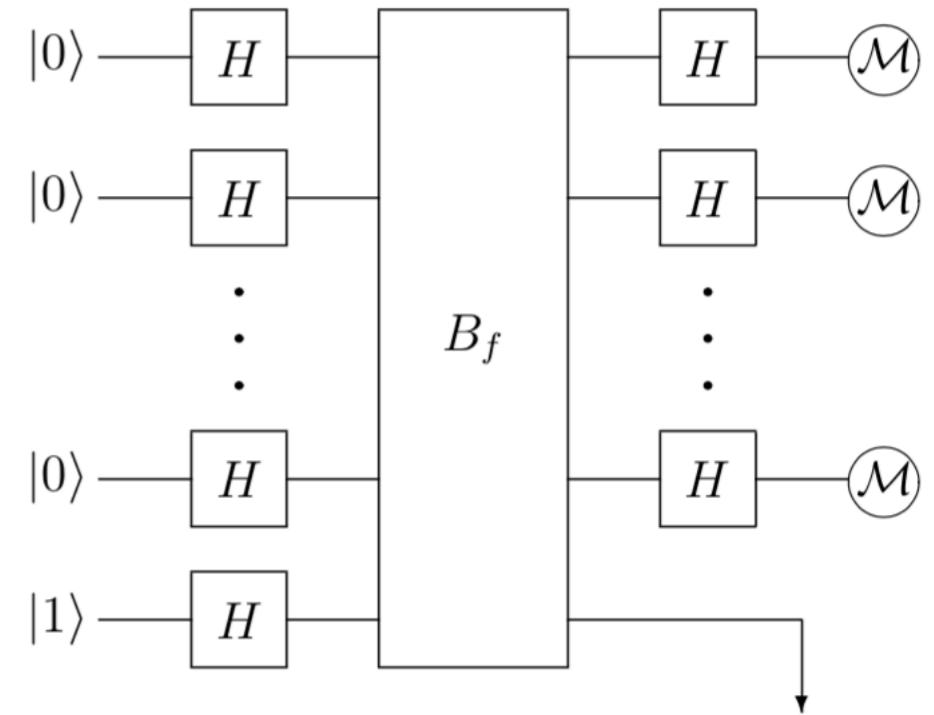
Algorithms

- Common theme of many quantum algorithms
 - Compute over the state space in superposition, essentially doing work “in parallel”
 - measure to produce the final result (and cancel out noise)
- Several challenges
 - Algorithms are probabilistic – must ensure measurement likely to get the right answer



Algorithms

- Common theme of many quantum algorithms
 - Compute over the state space in superposition, essentially doing work “in parallel”
 - measure to produce the final result (and cancel out noise)
- Several challenges
 - Algorithms are probabilistic – must ensure measurement likely to get the right answer
 - Gates in a quantum circuit must be unitary, implying (e.g.,) reversibility – “ancillae” bits used to encode answers



Recall: Challenges

In general

- Algorithms hard to write; answer may be unknown
- Simulation is intractable (states exponential in # qubits)

Recall: Challenges

In general

- Algorithms hard to write; answer may be unknown
- Simulation is intractable (states exponential in # qubits)
- Breakpoints break things (opening the box kills the cat)

Noisy, Intermediate Scale Quantum (NISQ) Computing era
— Preskill

Recall: Challenges

In general

- Algorithms hard to write; answer may be unknown
- Simulation is intractable (states exponential in # qubits)
- Breakpoints break things (opening the box kills the cat)

In the near term

Noisy, Intermediate Scale Quantum (NISQ) Computing era
— Preskill

Recall: Challenges

In general

- Algorithms hard to write; answer may be unknown
- Simulation is intractable (states exponential in # qubits)
- Breakpoints break things (opening the box kills the cat)

In the near term

- Execution is expensive, and (highly) error prone

Noisy, Intermediate Scale Quantum (NISQ) Computing era
— Preskill

Recall: Challenges

In general

- Algorithms hard to write; answer may be unknown
- Simulation is intractable (states exponential in # qubits)
- Breakpoints break things (opening the box kills the cat)

In the near term

- Execution is expensive, and (highly) error prone
- Computing resources (e.g., qubits) are scarce

Noisy, Intermediate Scale Quantum (NISQ) Computing era
— Preskill

Limited Qubits, Connectivity

ibmq_16_melbourne v1.0.0

online
Queue: 1 runs

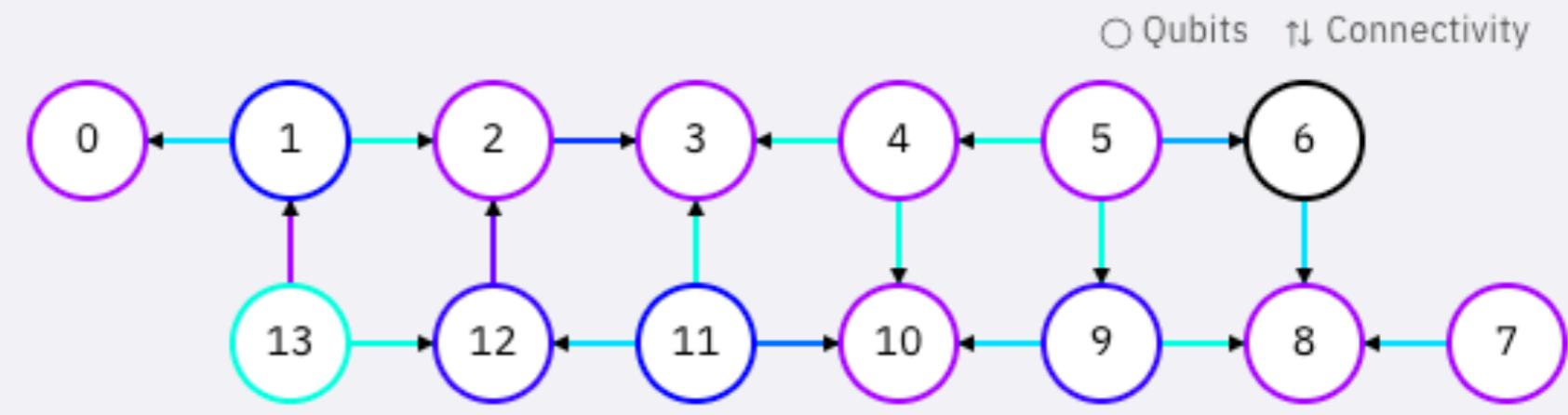
ibm-q/open/main

Accounts:

Hub: ibm-q

Group: open

Project: main



Single-qubit error rate

1.288e-2

CNOT error rate

3.845e-2

Qubits

14

Online since

2018-11-06

Basis gates

u1, u2, u3, cx, id

Limited Qubits, Connectivity

ibmq_16_melbourne v1.0.0

online
Queue: 1 runs

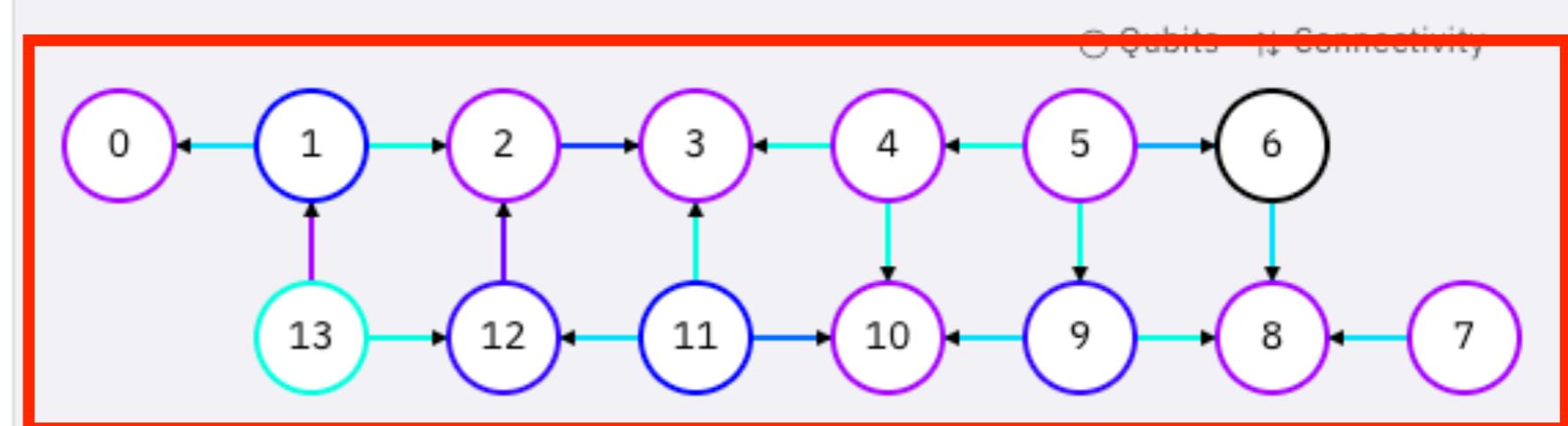
ibm-q/open/main

Accounts:

Hub: ibm-q

Group: open

Project: main



Single-qubit error rate

1.288e-2

CNOT error rate

3.845e-2

2.305e-2 1.525e-1

Qubits

14

Online since

2018-11-06

Basis gates

u1, u2, u3, cx, id

Limited Qubits, Connectivity

ibmq_16_melbourne v1.0.0

online
Queue: 1 runs

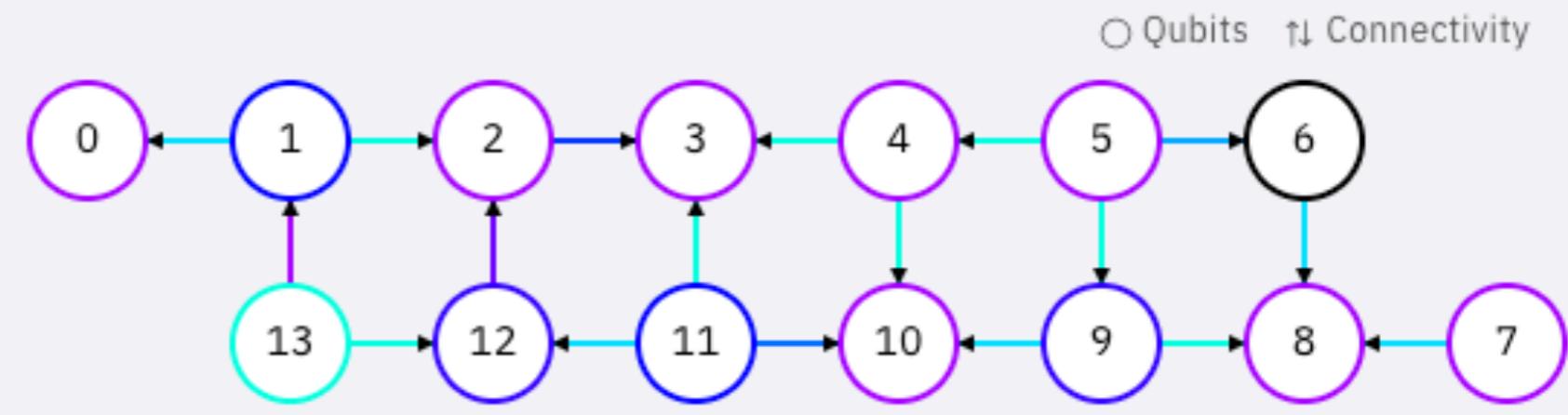
ibm-q/open/main

Accounts:

Hub: ibm-q

Group: open

Project: main



Single-qubit error rate

1.288e-2

CNOT error rate

3.845e-2

Qubits

14

Online since

2018-11-06

Basis gates

u1, u2, u3, cx, id

Limited Qubits, Connectivity

ibmq_16_melbourne v1.0.0

online
Queue: 1 runs

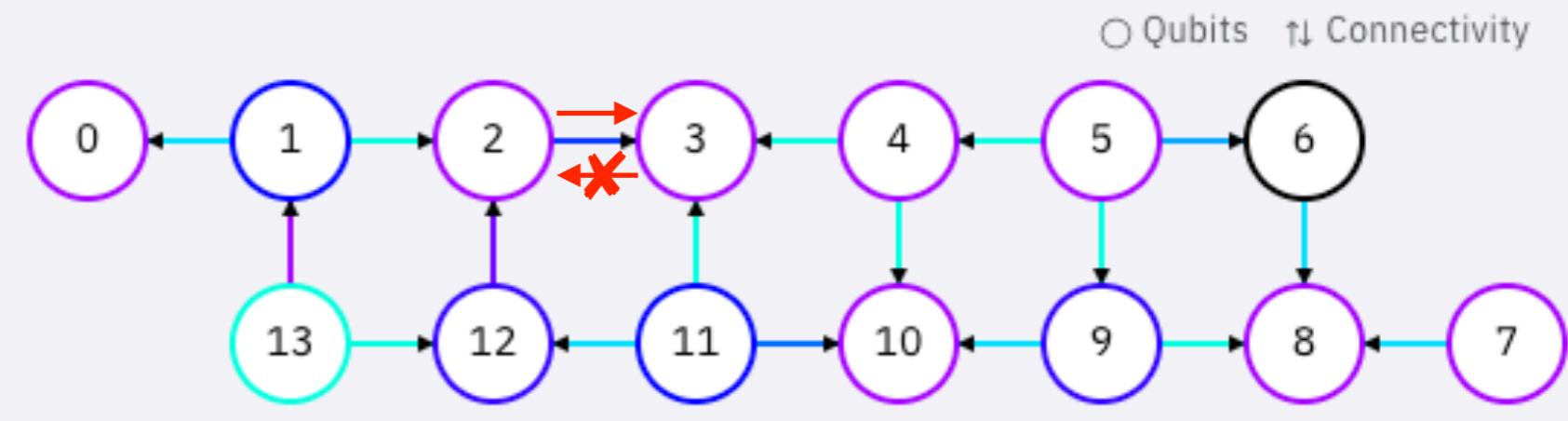
ibm-q/open/main

Accounts:

Hub: ibm-q

Group: open

Project: main



Single-qubit error rate

1.288e-2

CNOT error rate

3.845e-2

Qubits

14

Online since

2018-11-06

Basis gates

u1, u2, u3, cx, id

Limited Qubits, Connectivity

ibmq_16_melbourne v1.0.0

online
Queue: 1 runs

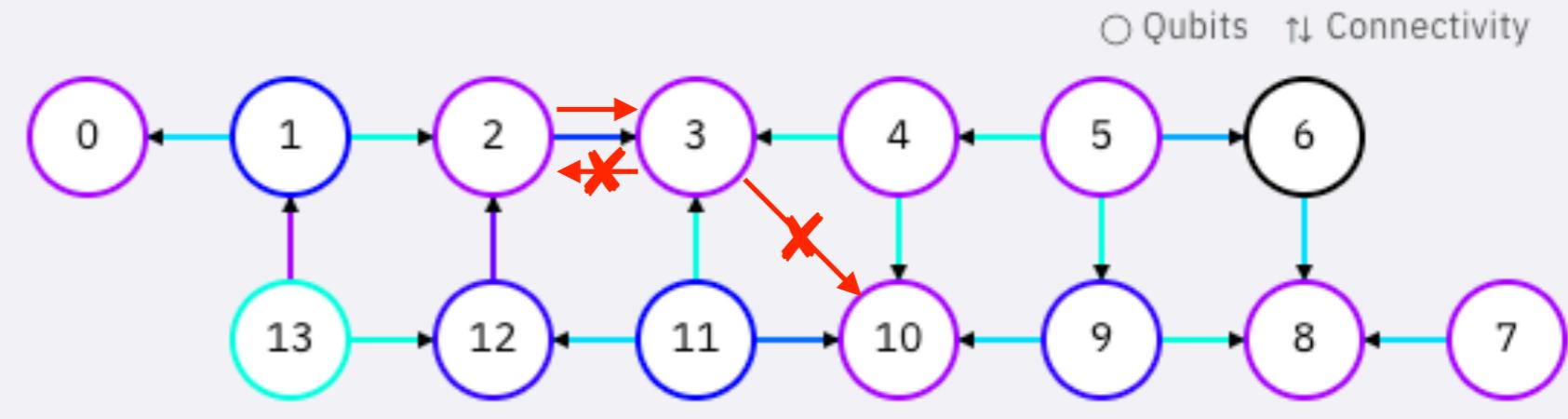
ibm-q/open/main

Accounts:

Hub: ibm-q

Group: open

Project: main



Single-qubit error rate

1.288e-2

2.305e-2

CNOT error rate

3.845e-2

1.525e-1

Qubits

14

Online since

2018-11-06

Basis gates

u1, u2, u3, cx, id

Substantial Error Rates

ibmq_16_melbourne v1.0.0

online
Queue: 1 runs

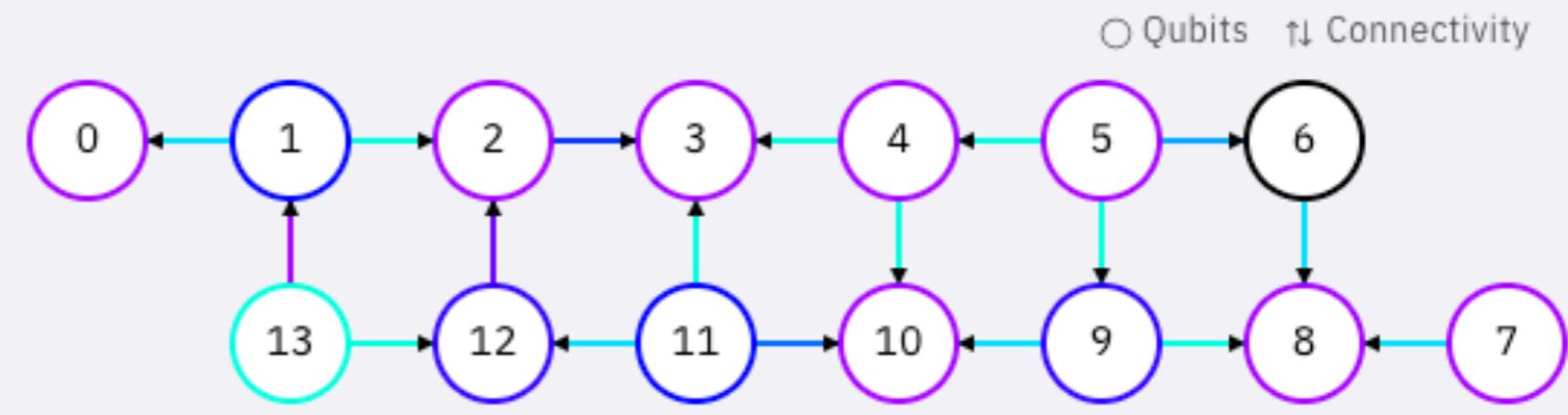
ibm-q/open/main

Accounts:

Hub: ibm-q

Group: open

Project: main

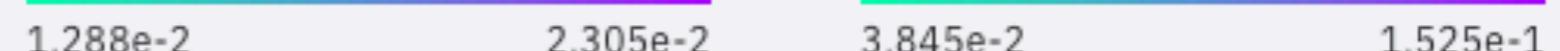


Single-qubit error rate

1.288e-2

CNOT error rate

3.845e-2



1.525e-1

Qubits

14

Online since

2018-11-06

Basis gates

u1, u2, u3, cx, id

Substantial Error Rates

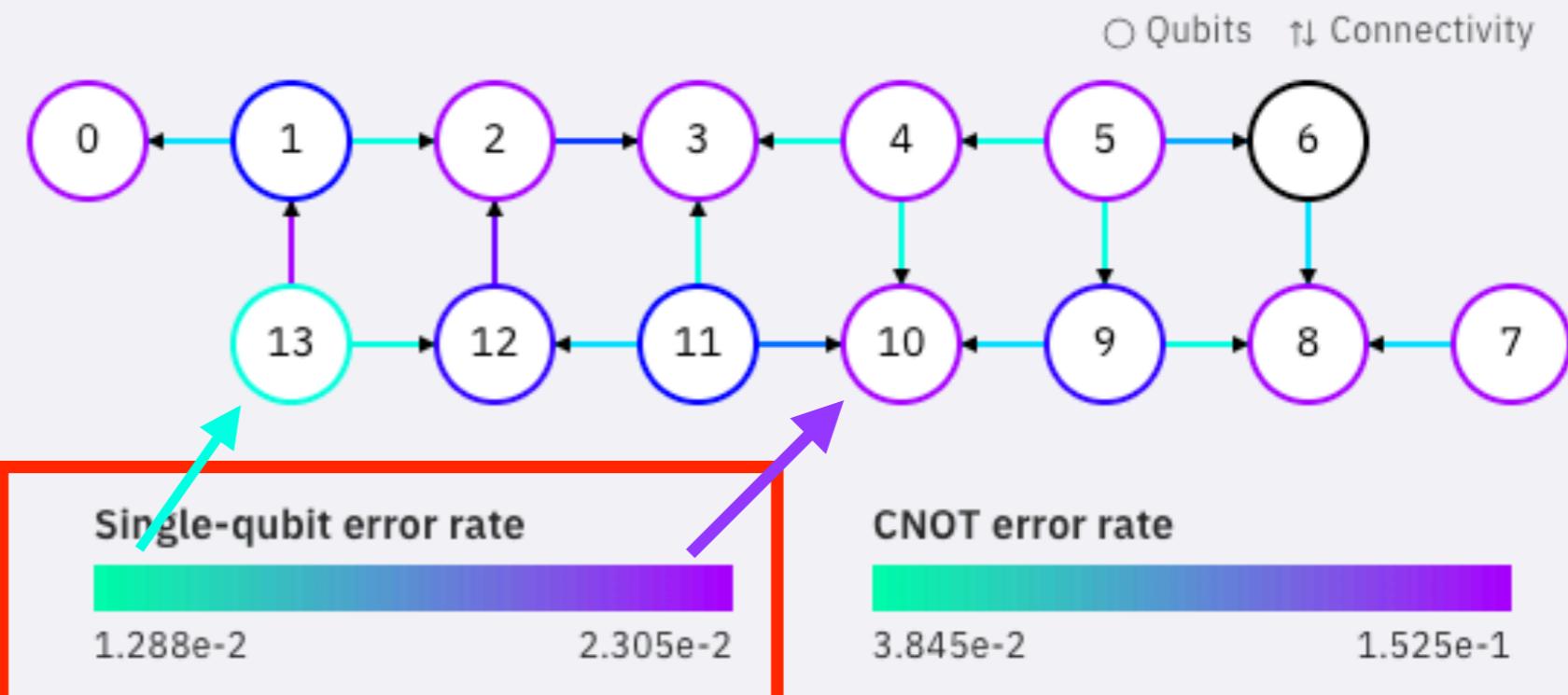
ibmq_16_melbourne v1.0.0

online
Queue: 1 runs

ibm-q/open/main

Accounts:

Hub: ibm-q
Group: open
Project: main



Qubits

14

Online since

2018-11-06

Basis gates

u1, u2, u3, cx, id

Substantial Error Rates

ibmq_16_melbourne v1.0.0

online

Queue: 1 runs

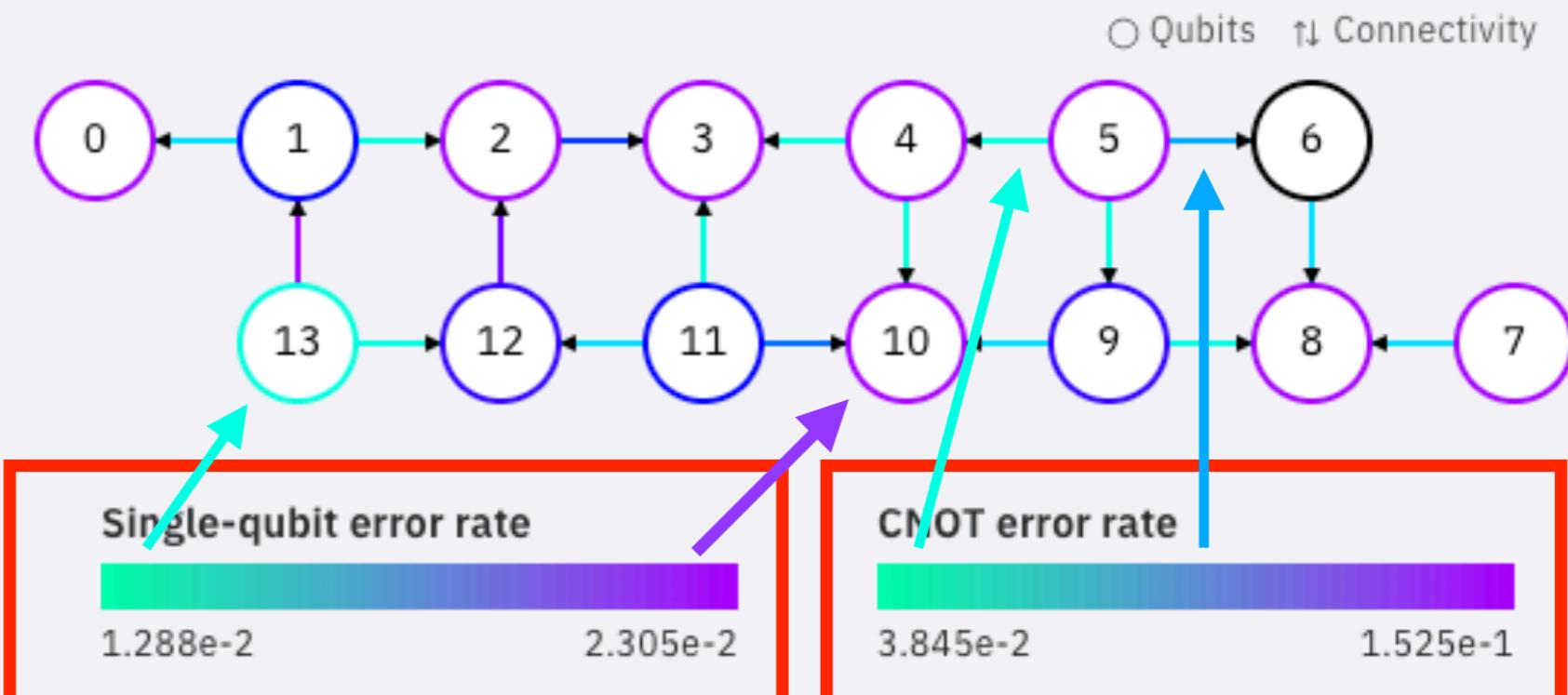
ibm-q/open/main

Accounts:

Hub: ibm-q

Group: open

Project: main



Qubits

14

Online since

2018-11-06

Basis gates

u1, u2, u3, cx, id

Addressing the Challenges

In general

Addressing the Challenges

In general

- **Formal verification:** State goal, prove you reach it
 - “Beware of bugs in the above code; I have only proved it correct, not tried it.” —Knuth

Addressing the Challenges

In general

- **Formal verification:** State goal, prove you reach it
 - “Beware of bugs in the above code; I have only proved it correct, not tried it.” —Knuth
- **Language design:** Make algorithms more intuitive

Addressing the Challenges

In general

- **Formal verification:** State goal, prove you reach it
 - “Beware of bugs in the above code; I have only proved it correct, not tried it.” —Knuth
- **Language design:** Make algorithms more intuitive

In the near term (handling scarce, error-prone resources)

- **Compiler optimizations** (proved correct)

Addressing the Challenges

In general

- **Formal verification:** State goal, prove you reach it
 - “Beware of bugs in the above code; I have only proved it correct, not tried it.” —Knuth
- **Language design:** Make algorithms more intuitive

In the near term (handling scarce, error-prone resources)

- **Compiler optimizations** (proved correct)
- **Reason about errors**, at the level of program and compiler

Quantum PL

**Verification
*Abstractions?***

**Compilation
Error Models
Verification
Optimization**

**Semantics
Mapping**

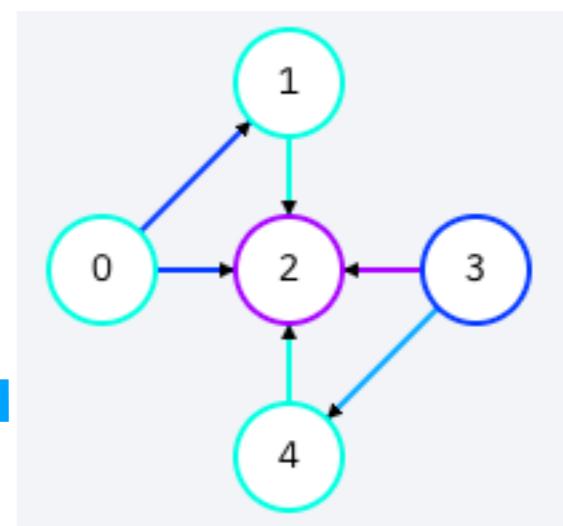
High Level Language

Intermediate (Circuit) Representation

Instruction Set

Hardware

**Hardware
Description,
Error Model**



Our Research

**Verification
*Abstractions?***

**Compilation
Error Models
Verification
Optimization**

**Semantics
Mapping**

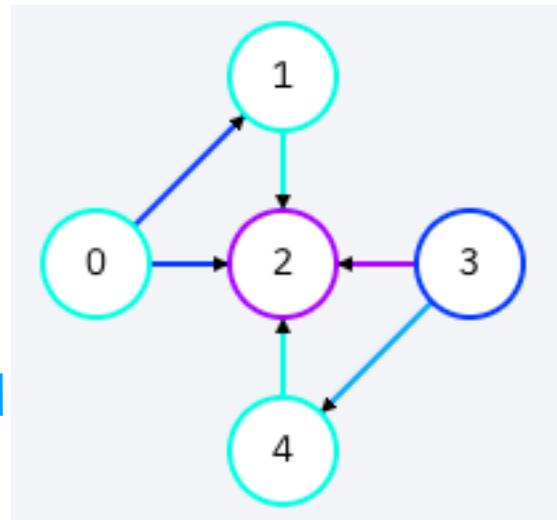
QWIRE, Robustness logic

SQIRE

(OpenQASM)

(IBM QX, Rigetti, IonQ)

**Hardware
Description,
Error Model**



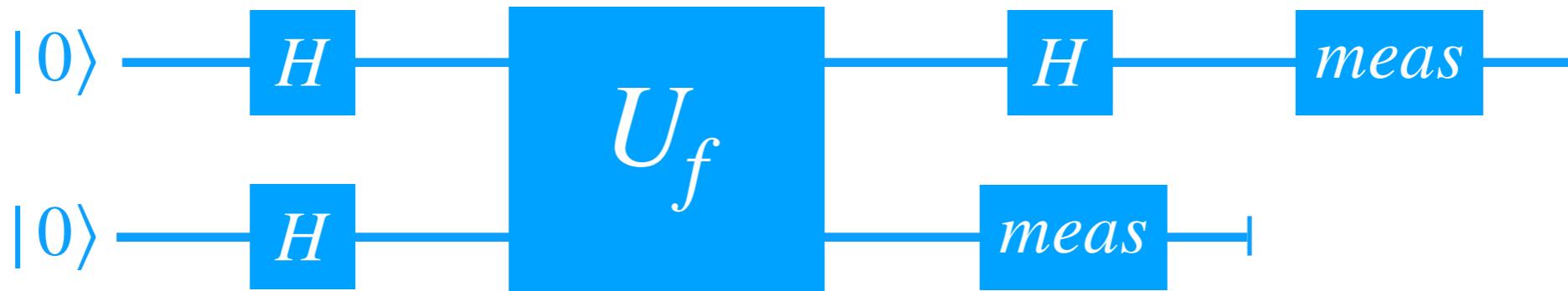
QWIRE

- QWIRE (POPL 2017) is a small language for describing quantum circuits. Features
 - A linear type system for enforcing *no cloning*
 - A denotational semantics in terms of *density matrices*
 - Embedded in Coq, with tactics for *program verification*

<https://github.com/inQWIRE/QWIRE>



Deutsch's Algorithm



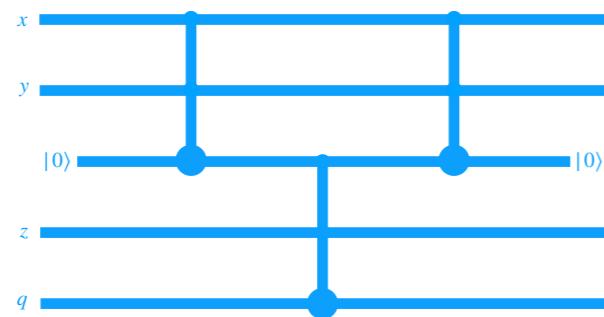
```
Definition deutsch (f : bool -> bool) :=
let x      ← H $ init0 $ ();
let y      ← H $ init1 $ ();
let (x,y) ← (U f) $ (x,y);
let ()    ← discard $ meas $ y;
meas $ H $ x.
```

QWIRE in action

QWIRE in action

Compilation

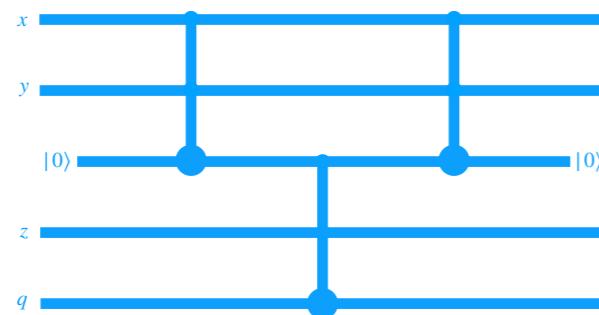
$$(x \wedge y) \wedge z$$



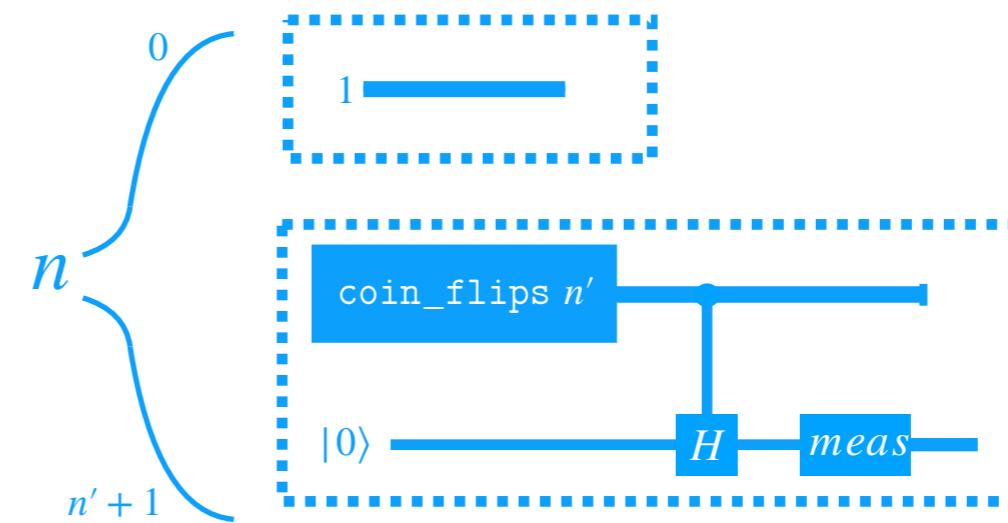
QWIRE in action

Compilation

$$(x \wedge y) \wedge z$$



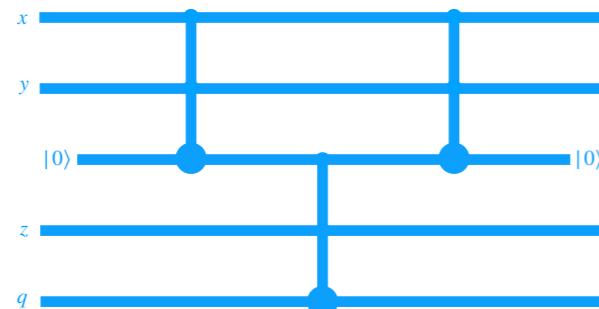
Random Number Generation



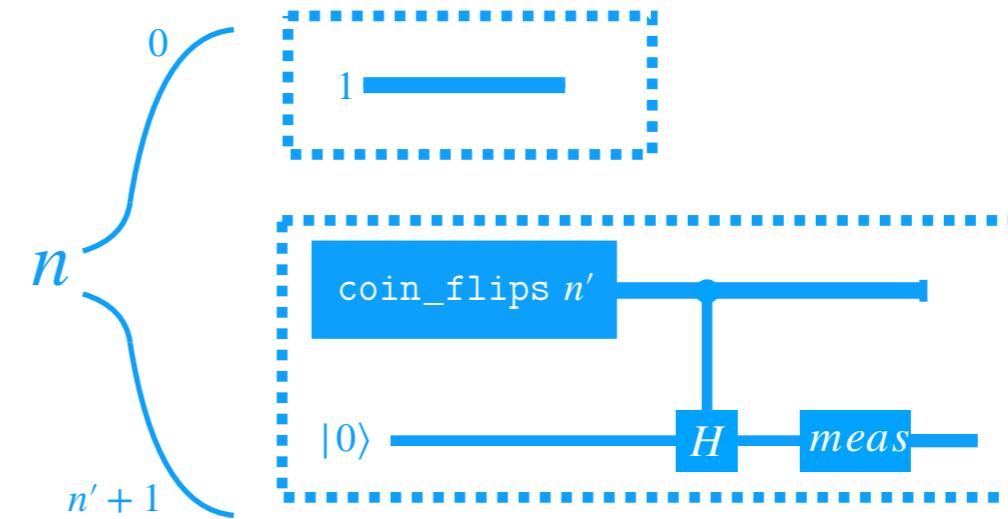
QWIRE in action

Compilation

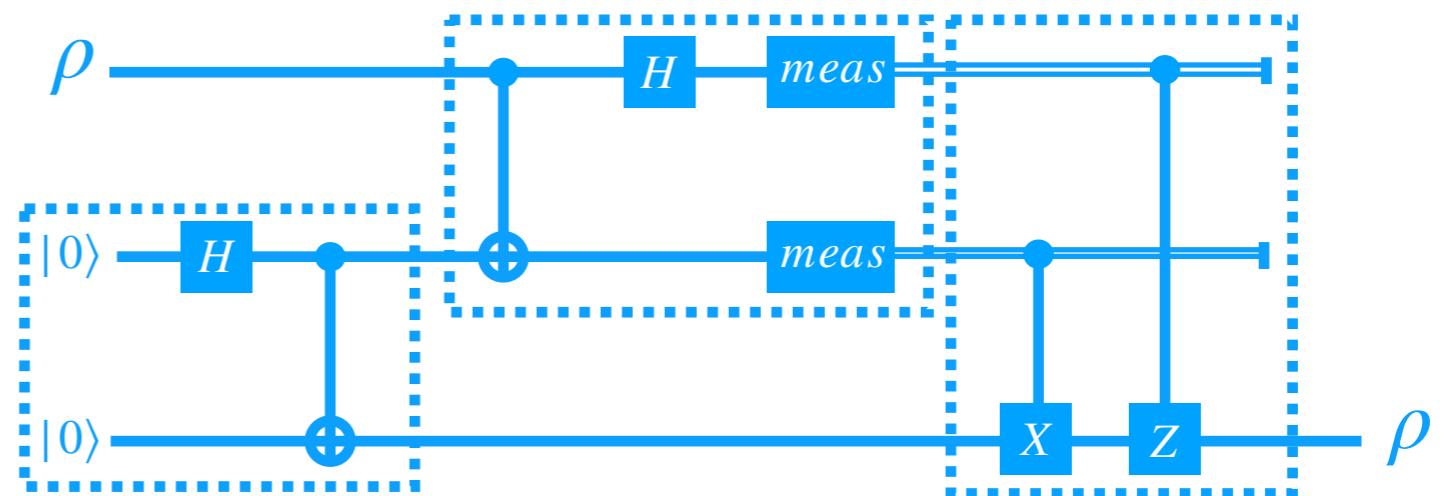
$$(x \wedge y) \wedge z$$



Random Number Generation



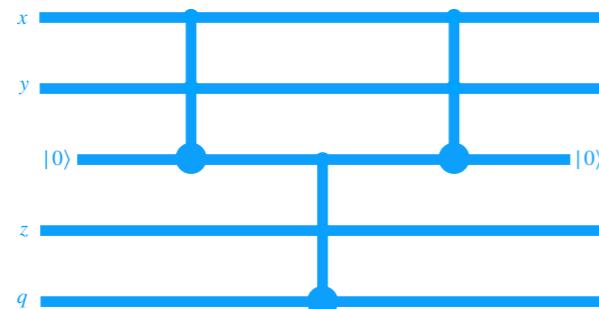
Teleportation



QWIRE in action

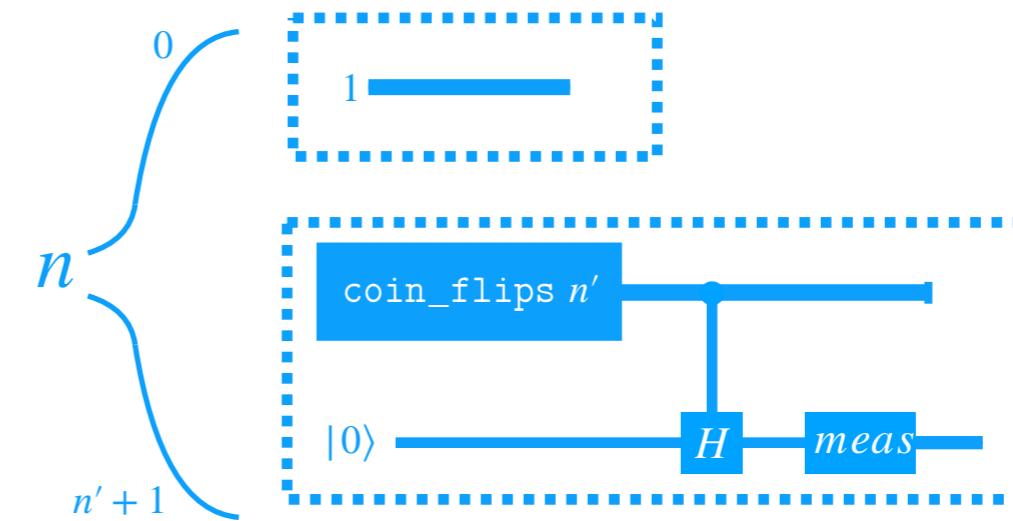
Compilation

$$(x \wedge y) \wedge z$$

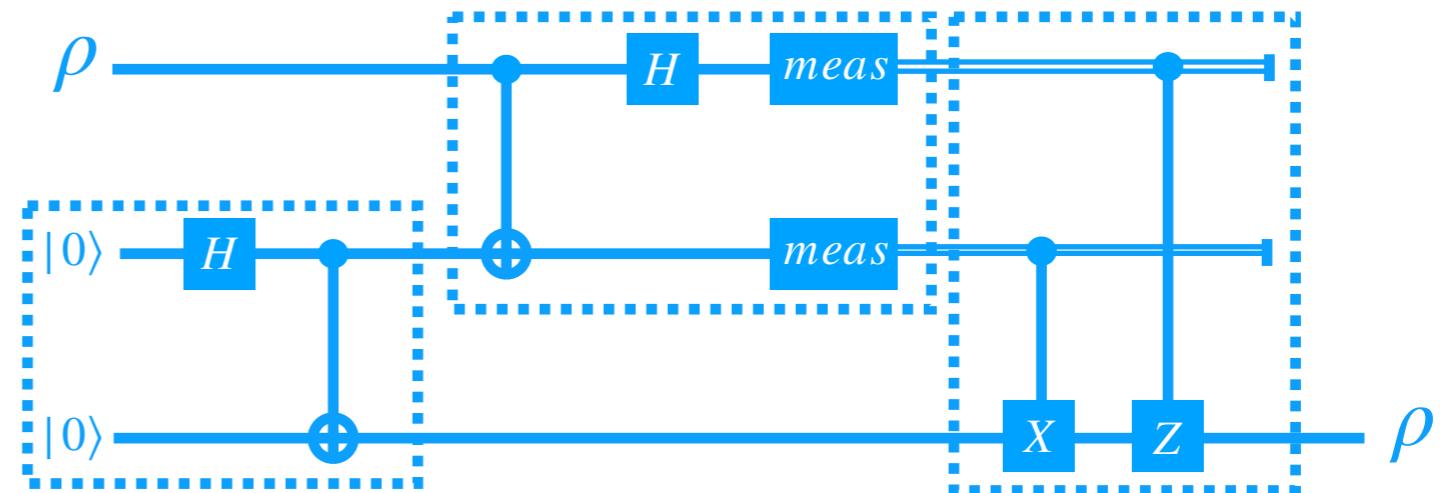


But: Nuts and bolts of proving things is surprisingly difficult!

Random Number Generation



Teleportation



Reasoning about Errors

- *Quantum Robustness Logic* (POPL 2019) allows us to bound errors in quantum programs given an error model on gates

$(Q, \lambda) \vdash \tilde{P} \leq \epsilon$.
 judgment bounds
 the “distance” ϵ of
 the actual state, due
 to errors, from the
 ideal

$$\begin{array}{c}
 \frac{}{(Q, \lambda) \vdash \text{skip} \leq 0} \text{ (Skip)} \quad \frac{}{(Q, \lambda) \vdash (q := |0\rangle) \leq 0} \text{ (Init)} \quad \frac{\|U \circ U^\dagger - \Phi\|_{Q, \lambda} \leq \epsilon}{(Q, \lambda) \vdash (\bar{q} \stackrel{\cong_{p, \Phi}}{=} U[\bar{q}]) \leq p\epsilon} \text{ (Unitary)} \\
 \frac{(Q', \lambda') \vdash \bar{P} \leq \epsilon' \quad \epsilon' \leq \epsilon \quad Q \sqsubseteq Q' \quad \lambda' \leq \lambda}{(Q, \lambda) \vdash \bar{P} \leq \epsilon} \text{ (Weaken)} \\
 \frac{(Q/\delta, \lambda/\delta) \vdash \bar{P} \leq \epsilon \quad 0 \sqsubseteq Q, Q/\delta \sqsubseteq I \quad 0 \leq \lambda, \lambda/\delta \leq 1}{(Q, \lambda) \vdash \bar{P} \leq \epsilon} \text{ (Rescale)} \\
 \frac{(Q_1, \lambda) \vdash \widetilde{P}_1 \leq \epsilon_1 \quad (Q_2, \lambda) \vdash \widetilde{P}_2 \leq \epsilon_2 \quad \{Q_1\}P_1\{Q_2\}}{(Q_1, \lambda) \vdash (\widetilde{P}_1; \widetilde{P}_2) \leq \epsilon_1 + \epsilon_2} \text{ (Sequence)} \\
 \frac{\forall m, (Q_m, 1 - \delta) \vdash \widetilde{P}_m \leq \epsilon \quad t, \delta \in [0, 1]}{(\sum_m M_m^\dagger Q_m M_m, 1 - t\delta) \vdash (\text{case } M[\bar{q}] = \overline{m \rightarrow \widetilde{P}_m} \text{ end}) \leq (1 - t)\epsilon + t} \text{ (Case)} \\
 \frac{(Q, \lambda) \vdash \widetilde{P}_1 \leq \epsilon \quad \{Q\}P_1\{\lambda M_0^\dagger M_0 + M_1^\dagger Q M_1\} \quad \widetilde{P} \equiv \text{while } M[\bar{q}] = 1 \text{ do } \widetilde{P}_1 \text{ done} \quad P \text{ is } (a, n)\text{-bounded}}{(\lambda M_0^\dagger M_0 + M_1^\dagger Q M_1, \lambda) \vdash \bar{P} \leq n\epsilon/(1 - a)} \text{ (While-Bounded)} \\
 \frac{}{(Q, \lambda) \vdash (\text{while } M[\bar{q}] = 1 \text{ do } \widetilde{P}_1 \text{ done}) \leq 1} \text{ (While-Unbounded)}
 \end{array}$$

Reasoning about Errors

- *Quantum Robustness Logic* (POPL 2019) allows us to bound errors in quantum programs given an error model on gates. Limitations:

Reasoning about Errors

- *Quantum Robustness Logic* (POPL 2019) allows us to bound errors in quantum programs given an error model on gates. Limitations:
 - These bounds only *increase*: No way to verify error correction.

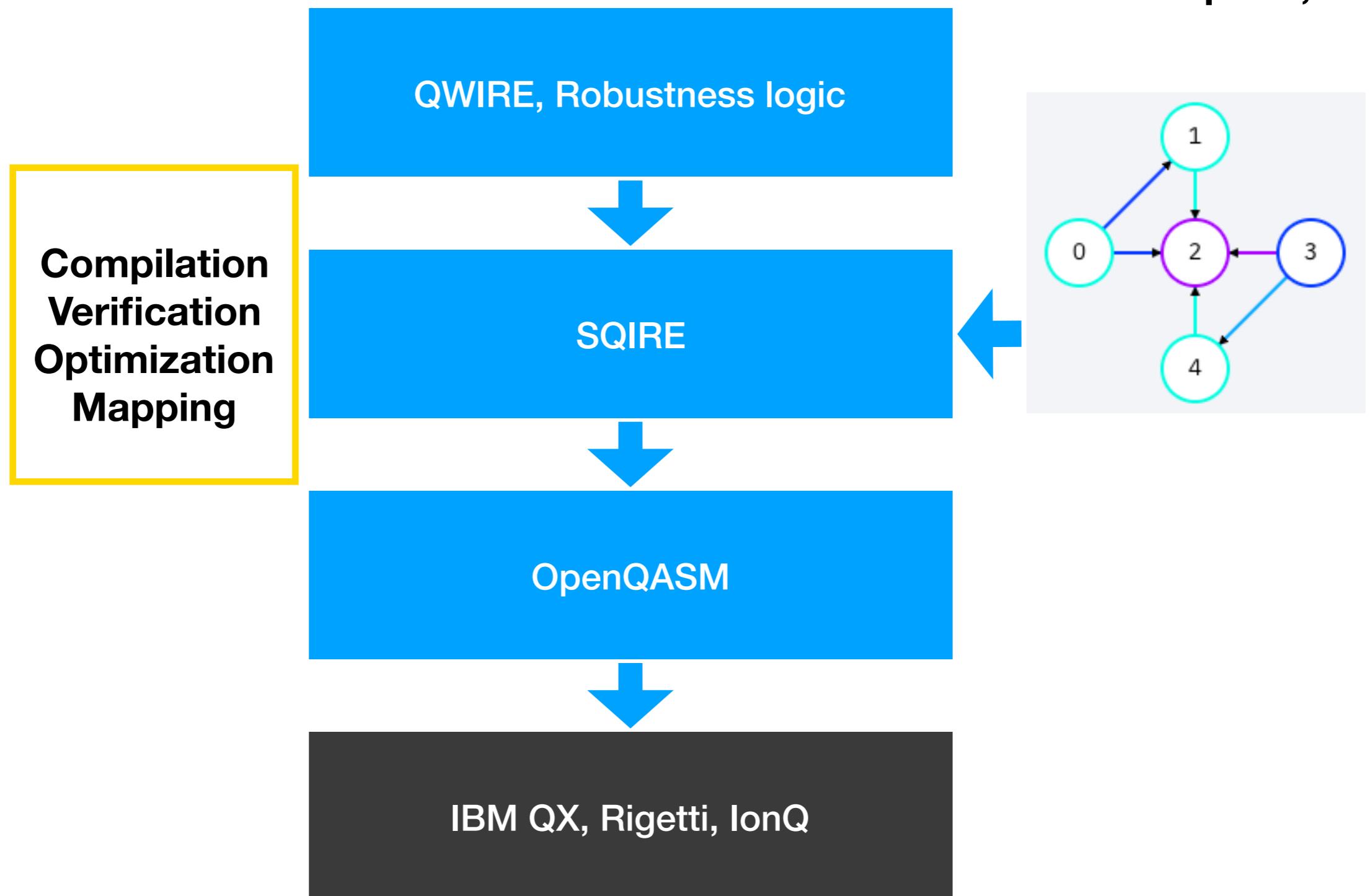
Reasoning about Errors

- *Quantum Robustness Logic* (POPL 2019) allows us to bound errors in quantum programs given an error model on gates. Limitations:
 - These bounds only *increase*: No way to verify error correction.
 - Not designed for general reasoning

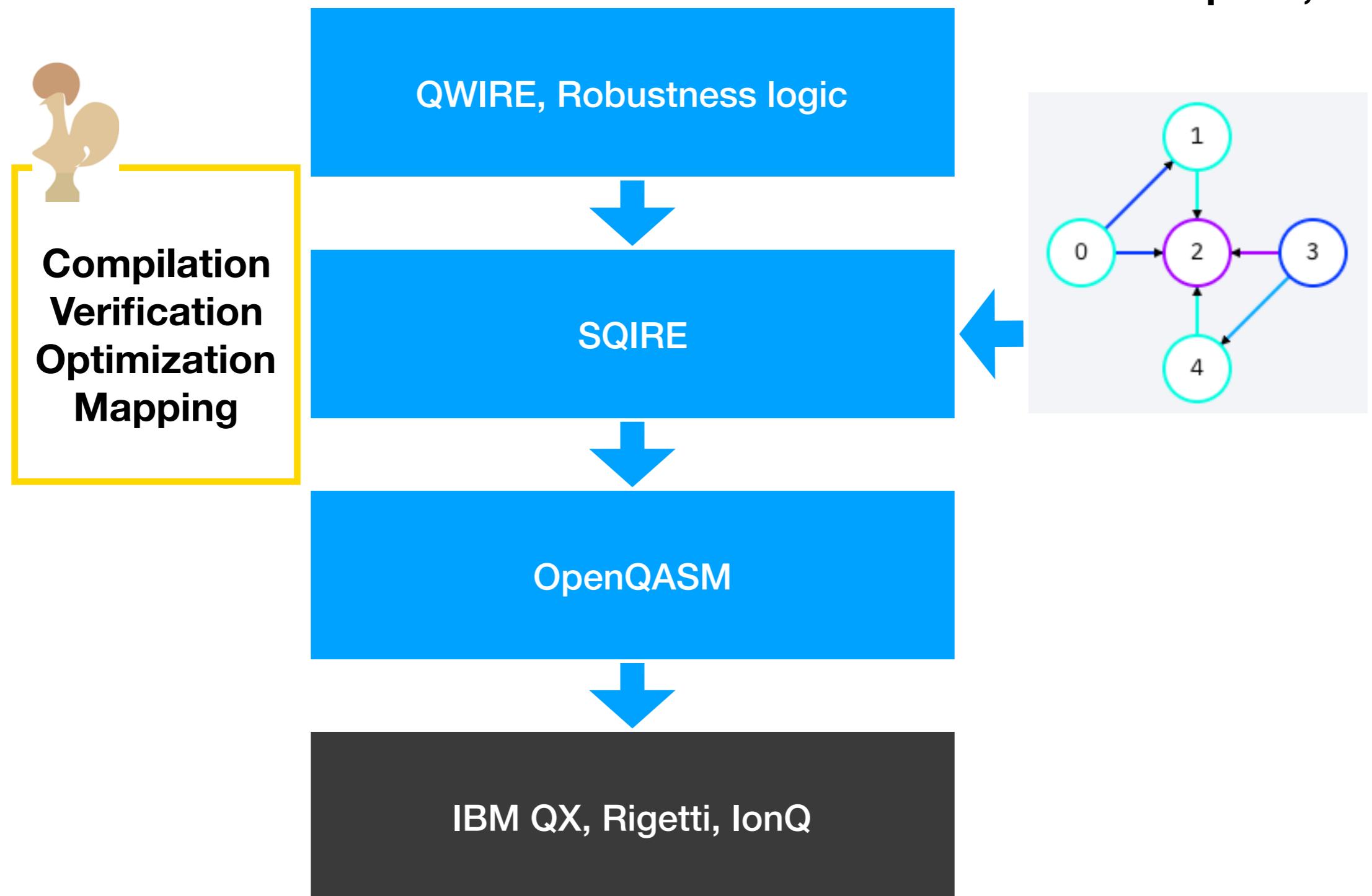
Reasoning about Errors

- *Quantum Robustness Logic* (POPL 2019) allows us to bound errors in quantum programs given an error model on gates. Limitations:
 - These bounds only *increase*: No way to verify error correction.
 - Not designed for general reasoning
 - Not tied to hardware

Our Research



Our Research



SQIRe

- Simple IR for certified quantum compiler
 - Past compilers have had bugs. In SQIRe, can prove optimizations are bug free
 - Likewise, prove that mapping to hardware with limits makes sense
- Can also prove algorithm-level properties. Use of global register, rather than HOAS, a key
 - But not sure if key in principle, or just our practice
- Goal: Error-aware optimizations

Horizon: Higher-level Languages

- Current “high level” languages not very high level.
 - Traditional control operators over quantum circuits
- Instead: we want higher-level building blocks
 - Expose key components of quantum algorithms
 - Q# is pushing in this direction, but still much to do

Future Work: You!

- There is lots of room for work in quantum computing that takes a PL perspective
 - Formal methods, compilers & optimization, language design, static analysis, hardware/software co-design ...
- Join us!

