SAND2017-1508C

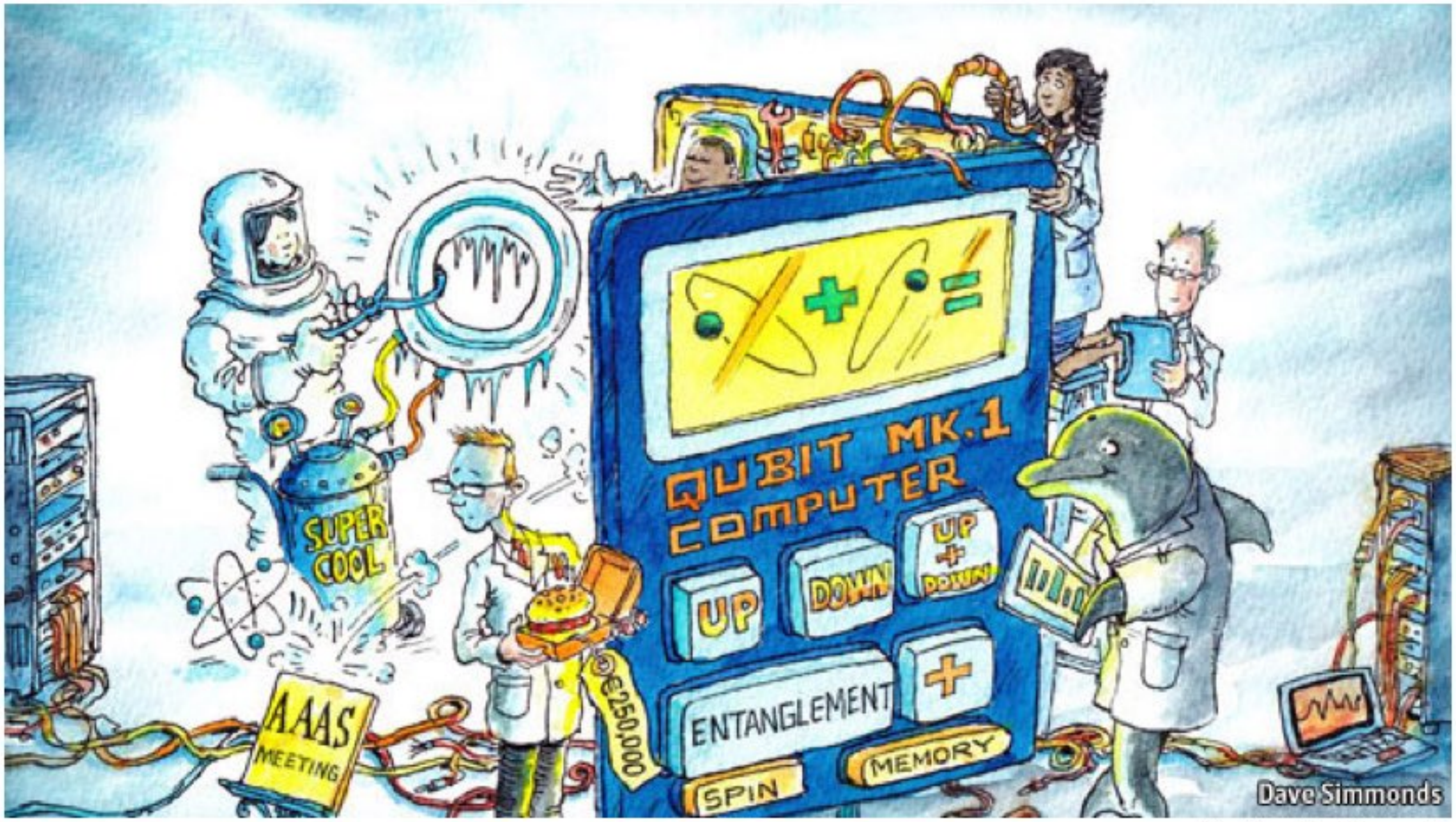
Testbeds as Pathfinders: Probing Noise

Robin Blume-Kohout

Center for Computing Research
Sandia National Labs

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

Why build testbeds?



Credit: The Economist

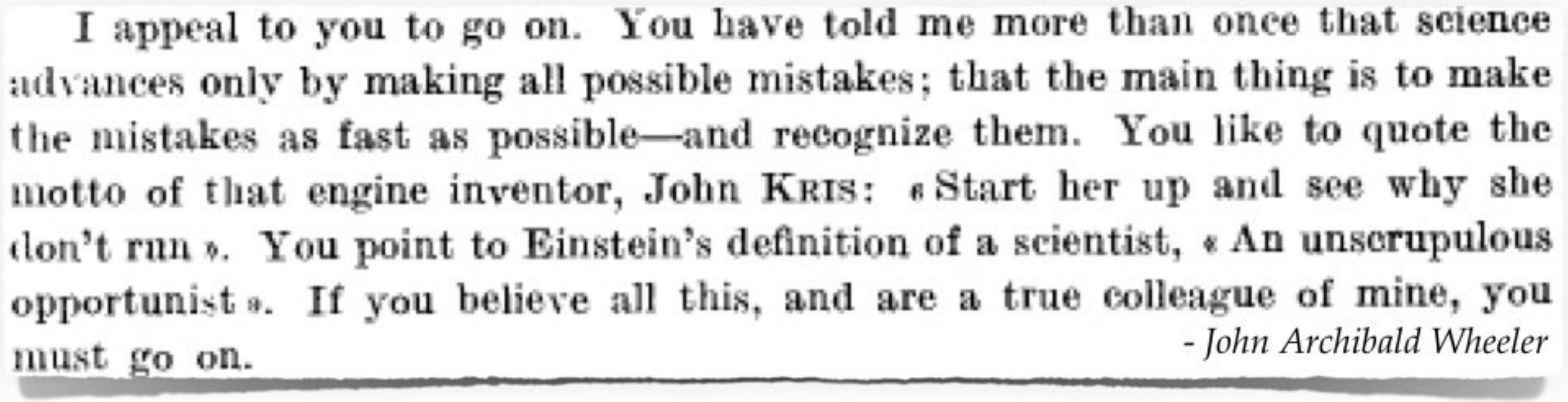
Will it work?

(What does "work" mean?)

How do we know if it works?

(How do we make it better?)

Why build testbeds?



I appeal to you to go on. You have told me more than once that science advances only by making all possible mistakes; that the main thing is to make the mistakes as fast as possible—and recognize them. You like to quote the motto of that engine inventor, John KRIS: « Start her up and see why she don't run ». You point to Einstein's definition of a scientist, « An unscrupulous opportunist ». If you believe all this, and are a true colleague of mine, you must go on.

- John Archibald Wheeler

Why build testbeds?

Start 'er up
and see why
she don't run.

I. **What we'll learn from testbeds**

II. **How we will study testbeds**

III. **Control & interface requirements**

What we need to learn from testbeds

5-15 qubit QIPs are a *new* experimental system, with *new failure modes* to study.

Simulating quantum circuits with *realistic errors* on $>$10 qubit QIPs is very hard.

- We don't know *what kind of noise and errors* afflict 5-15 qubit QIPs.
  We need hard data on what kinds of failures happen.
- We don't know how *realistic errors affect QEC circuits.*
  We need to run <u>QEC circuits</u> with realistic, tunable errors.
- We don't know how *algorithms degrade because of realistic errors.*
  We need to run <u>algorithms</u> with realistic, tunable errors.
- We don't know how to tune and debug 5-15 qubit QIPs.
  We need to test our characterization and tuning methods.
- Summary: testbeds are *data sources* and *pathfinders.*

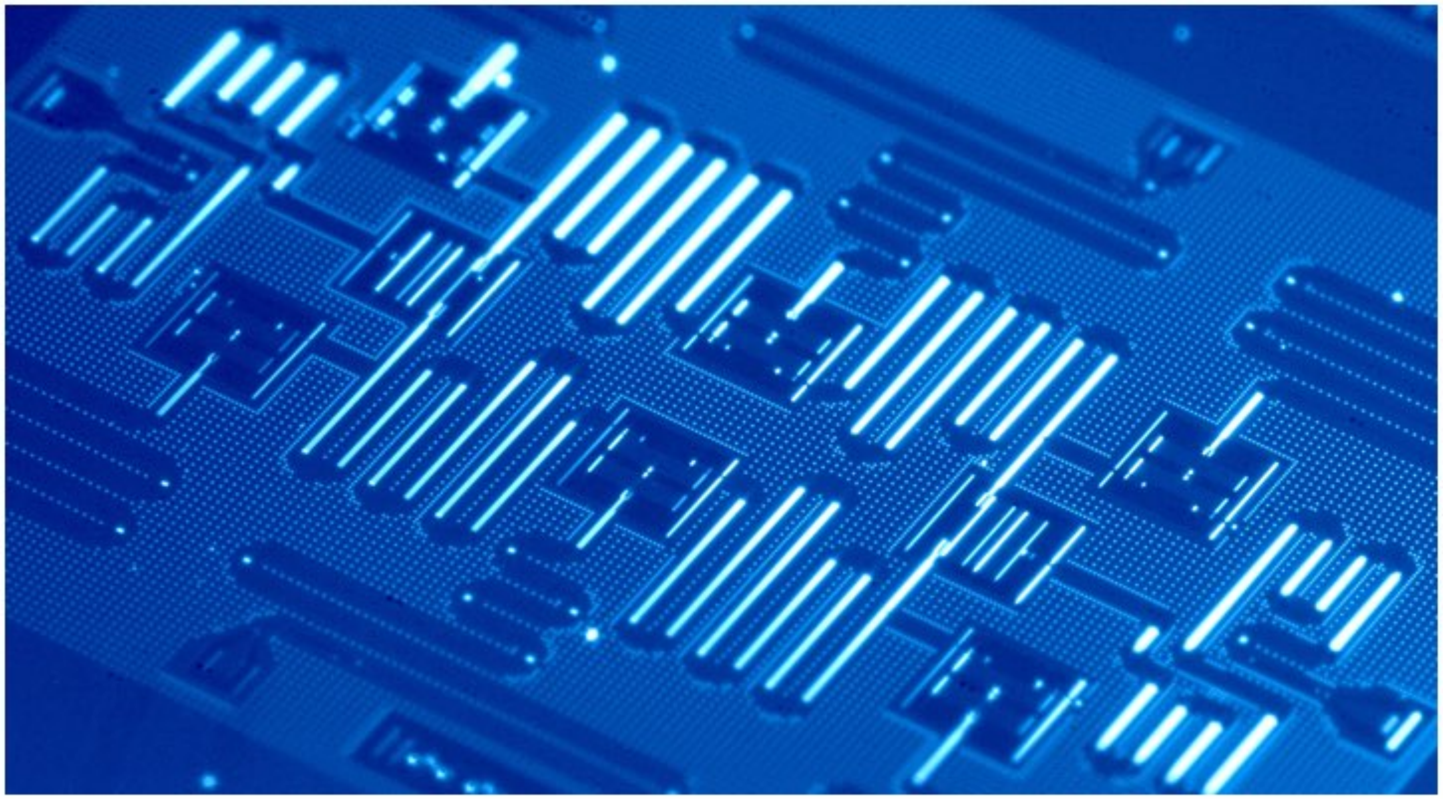
I. What we'll learn from testbeds

II. How we will study testbeds

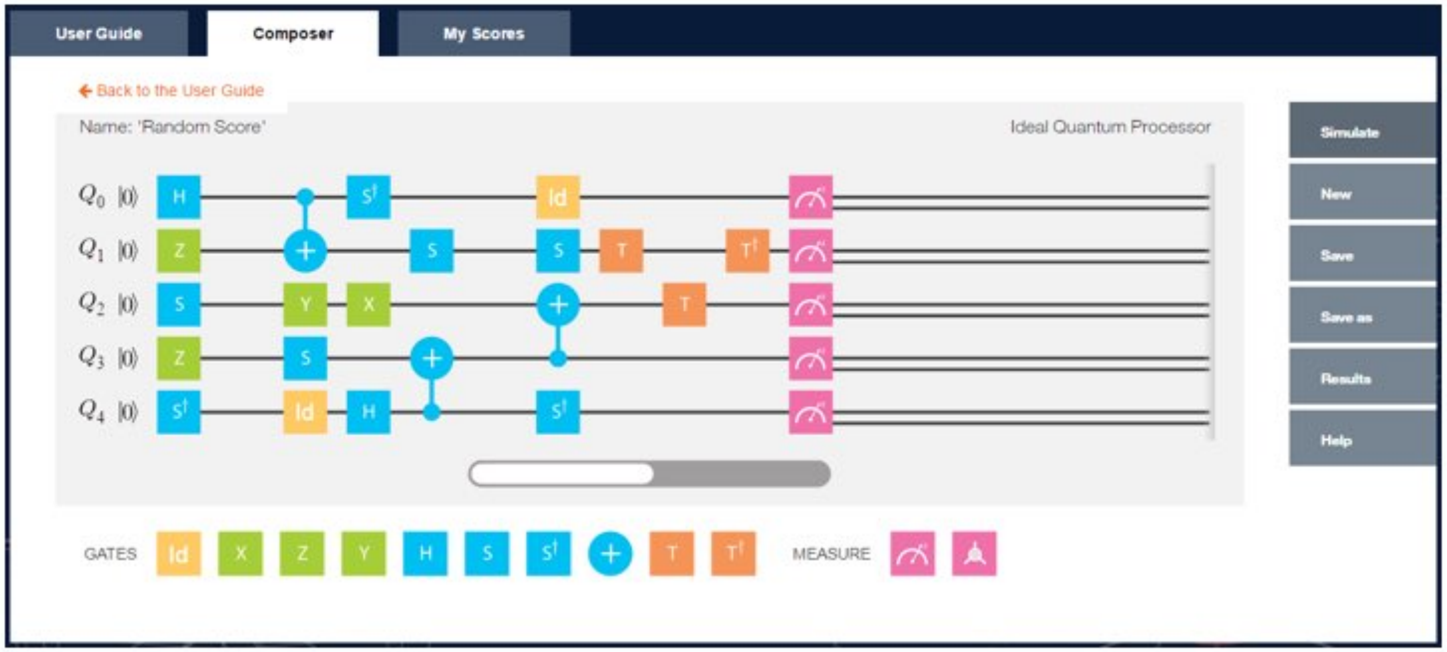
III. Control & interface requirements

Photo Credit: IBM Quantum Experience
(publicly released)

HARDWARE

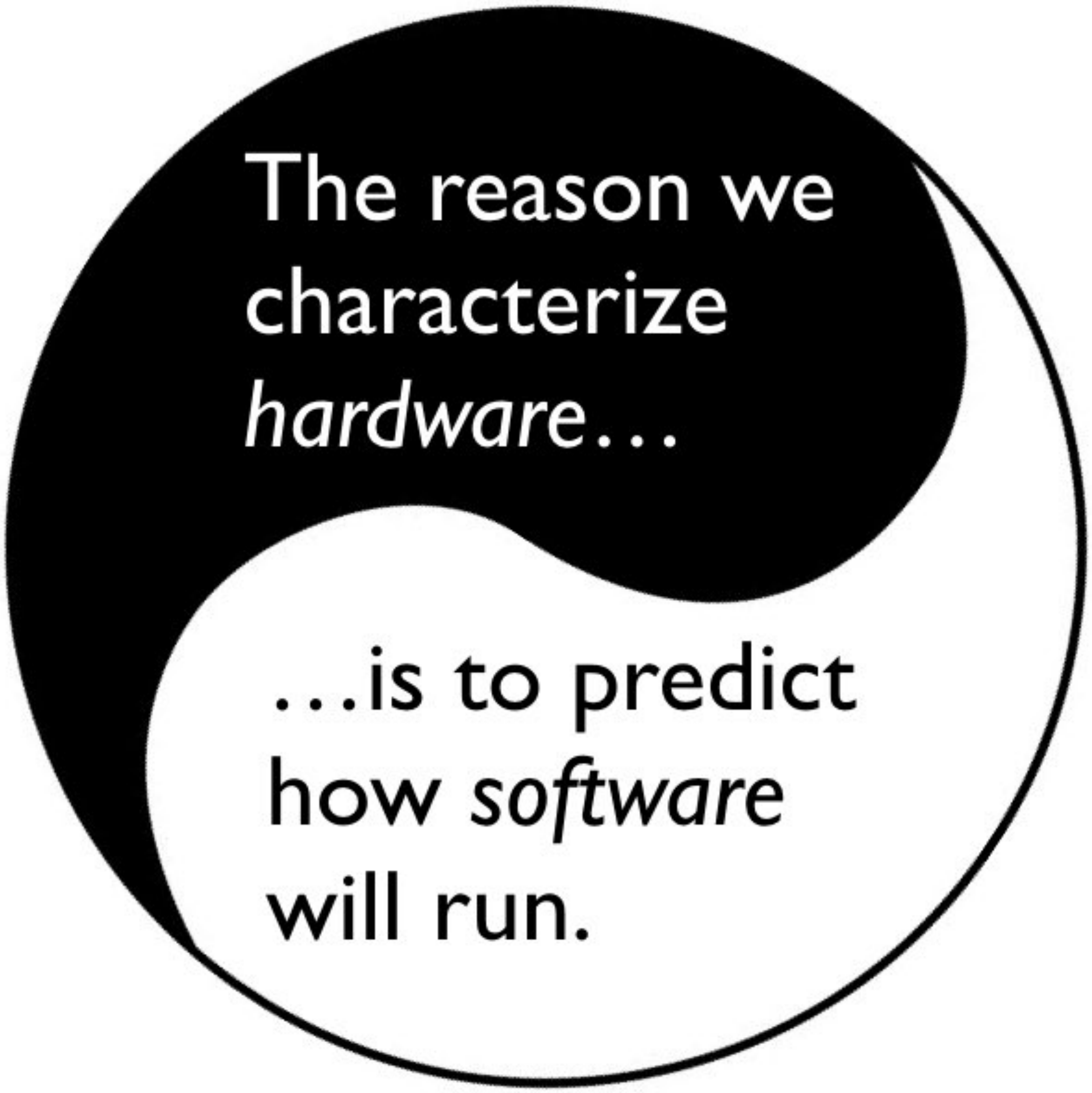


SOFTWARE



Characterizing *hardware*…

…is a *software* application.



The reason we characterize *hardware*…

…is to predict how *software* will run.

Modern Characterization Methods

Old-school QCVV

State tomography
Process tomography
Ancilla-assisted tomography
Direct fidelity estimation
...

- Rely on unprovable assumptions about QIP control.
- Don't respect *gauge symmetry.*
- Not built on *quantum circuits.*
- In general, **not fully reliable**.

Black-box methods

These methods treat the QIP as a black box with *strictly classical control and measurement*, and try to assume as little as possible.

Randomized Methods

Example: ***Randomized Benchmarking (RB)***

Emphasize simplicity, sacrifice detail.

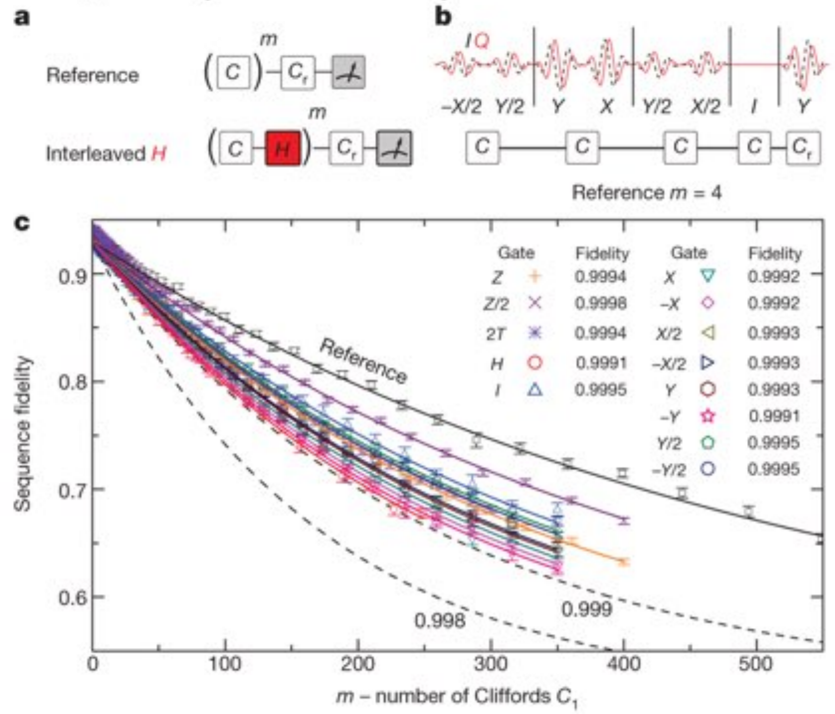
Model-based Methods

Example: ***Gate-Set Tomography (GST)***

Emphasize predictive detail, sacrifice simplicity.

RB: State of play (2016)

- Dominant characterization method since 2007 for 1-2 qubit processors.
- Scales pretty well with N_{qubits}.
- Only reports <u>one number</u> to describe the entire QIP probed.
- Variants have arisen over the past 3-5 years to probe more numbers.
- Not intended/suitable for *predictive modeling* — you can do it, but only using strong and unlikely assumptions.

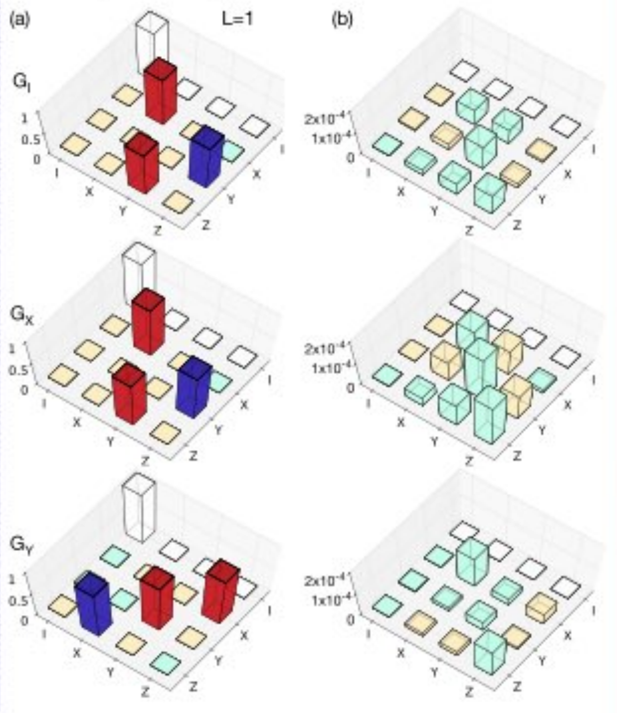


Barends et al, *Nature* **508**, 500–503 (24 April 2014)

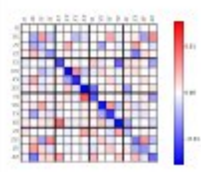
GST: State of play (2016)

- Proposed by IBM in 2013,
  Extended and developed by Sandia,
  Used by 10-20 experimental groups.
- Open-source *PyGSTi* software:
  full characterization of 1-2 qubit gatesets.
- More complex than RB
  (100 - 35,000 different circuits).
- Will require breakthrough creativity
  to scale beyond 2 qubits.

1-qubit gateset estimates



2-qubit GST gate estimate

Gate	Error Generator	Hamiltonian Projections	Stochastic Projections
Gix		$\begin{array}{c\|cccc} \text{Q1 \textbackslash Q2} & I & X & Y & Z \\ \hline I & 0 & -0 & -.01 & -.02 \\ X & .01 & -0 & -0 & .01 \\ Y & 0 & 0 & -0 & -.01 \\ Z & -0 & 0 & .01 & .02 \end{array}$	$\begin{array}{c\|cccc} \text{Q1 \textbackslash Q2} & I & X & Y & Z \\ \hline I & -.04 & 0 & .01 & 0 \\ X & 0 & 0 & 0 & 0 \\ Y & 0 & 0 & 0 & 0 \\ Z & .02 & 0 & 0 & 0 \end{array}$

*Post*modern Characterization Methods

- Next 3 years: circuit-based black-box protocols that characterize
  - Leakage
  - Crosstalk
  - Drift (time-dependence)
  - Correlated errors
  - Large processors (holistically)

- Today, even in 1-2 qubit processors we *know* our models don't work. In 3 years, we hope to be correctly modeling all visible noise.

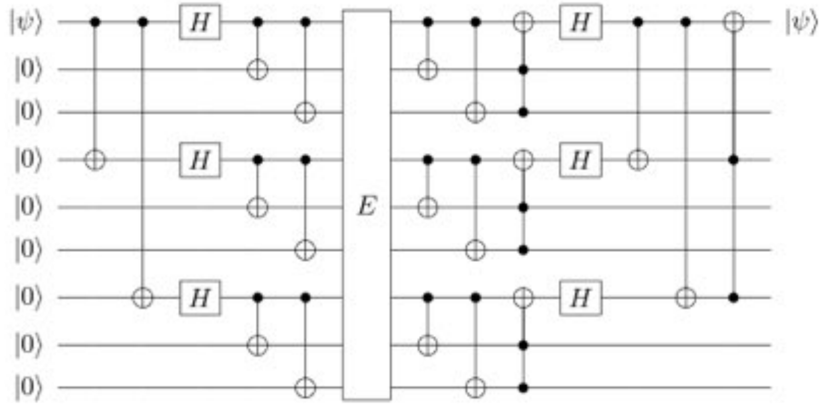
- More exciting: Future protocols will directly probe the effect of **changing how we implement gates**. This will be critical for *improving* our devices and *stabilizing* them against drift.

I. What we'll learn from testbeds

II. How we will study testbeds

III. Control & interface requirements

Control for *modern* QCVV



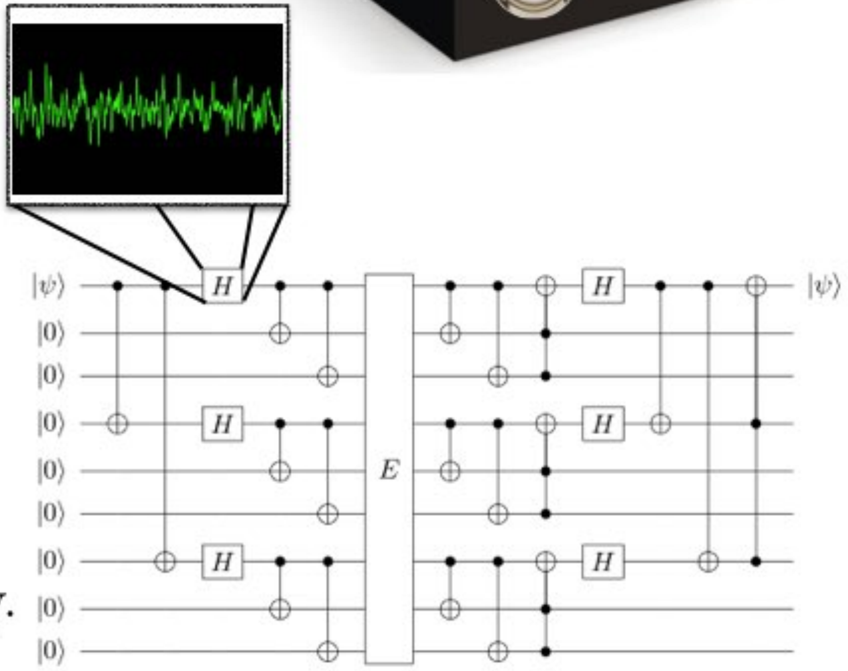
- Probing multiqubit QIPs requires: **running lots and lots of circuits**.
- Many qubits can fail in many ways. Characterizing to enable debugging, sophisticated QEC, and prediction requires lots of data — *taken as rapidly as possible* before drift.
- Control system: *hierarchical FPGAs* or *streaming ID* capability.
  ==> reprogram (change circuits) on the fly.
  ==> avoid limitations on flexibility caused by *AWG memory limits.*
- Question for discussion: do we need real-time feedback?
  - ability to change circuits in *near*-real time: definitely useful.
  - actual real-time feedback capability: maybe not necessary?

Control for *postmodern* QCVV

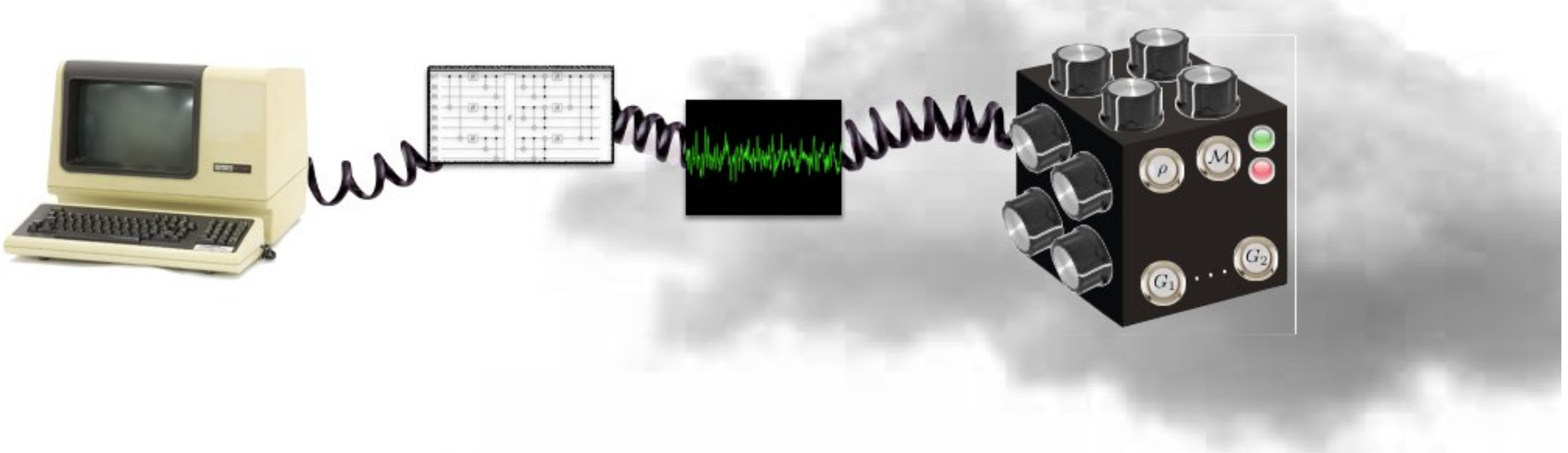
- Don't build your control system for 2017.
  Build it for the future!

- So what will *future* characterization look like?
  - the same, but more,
  - feedback control to counteract drift,
  - optimizing pulse shaping, gate design.

- What does this demand of control?
  - **well-defined ways to open up the black box.**
  - "knobs" to control analog params.
  - fast feedback for analog control.
  - a language (doesn't exist today) for letting *users* describe gate variations efficiently and powerfully.



Thoughts on Remote Access



- The Sandia QCVV group has a lot of experience with remote QCVV.
- We think "cloud" access will work *if* effort is put into *making* it work.
  - IBM's "Quantum Experience" is a useful pathfinder here.
  - Diverse users *must* be part of the interface design process, early!
  - Circuits are the base layer — what's the best language for them?
  - *Implementation transparency is critical*! No hidden optimization.
  - A language for remote *analog* controllability does not exist yet; defining one (flexible, usable, extensible) is a high priority.