International Symposium on Grids and Clouds 2024.03.26

Building a superconducting quantum computer

陳啟東, 中央研究院物理研究所

ChiiDong Chen, Institute of Physics, Academia Sinica

http://www.phys.sinica.edu.tw/~quela

https://www.nextplatform.com/2020/02/18/quantum-control-more-than-meets-the-eye/

Outline:

- Quantum Logic Gates and Quantum Computers: an Introduction
- Realization of a Superconducting Quantum Computer
- The need for a tight quantum-classical integration

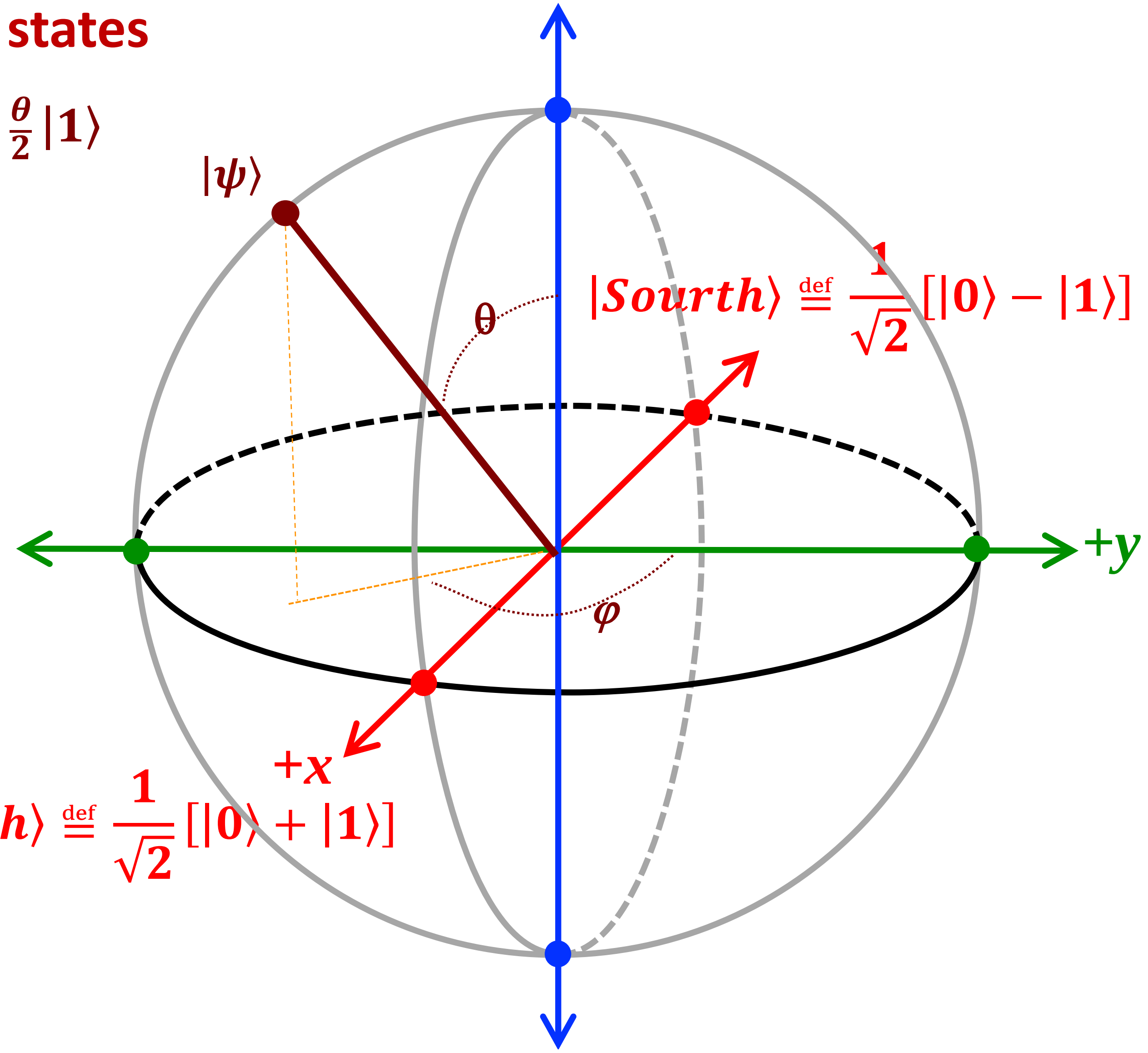
Bloch sphere:

Visualization of qubit states

$$|\psi\rangle = \cos\frac{\theta}{2}|0\rangle + e^{i\varphi}\sin\frac{\theta}{2}|1\rangle$$

$$= \alpha|0\rangle + \beta|1\rangle$$

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



$|g\rangle \stackrel{\text{def}}{=} |0\rangle = |north\ pole\rangle$

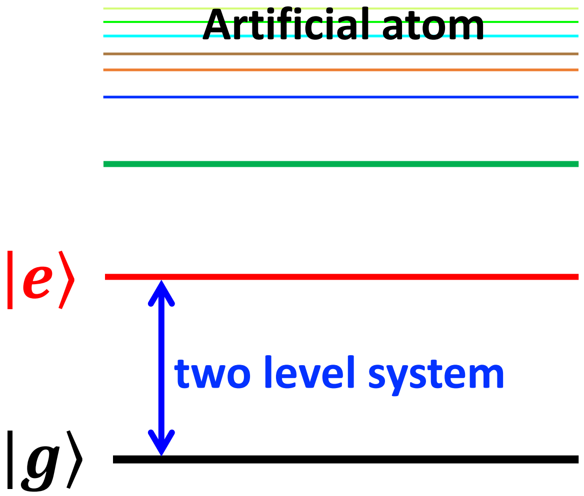
$|Sourth\rangle \stackrel{\text{def}}{=} \frac{1}{\sqrt{2}}[|0\rangle - |1\rangle]$

$|East\rangle \stackrel{\text{def}}{=} \frac{1}{\sqrt{2}}[|0\rangle - i|1\rangle]$

$|West\rangle \stackrel{\text{def}}{=} \frac{1}{\sqrt{2}}[|0\rangle + i|1\rangle]$

$|North\rangle \stackrel{\text{def}}{=} \frac{1}{\sqrt{2}}[|0\rangle + |1\rangle]$

$|e\rangle \stackrel{\text{def}}{=} |1\rangle = |sourth\ pole\rangle$



Artificial atom

$|e\rangle$

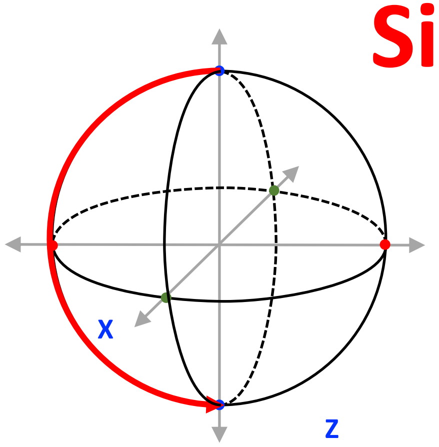
two level system

$|g\rangle$

Single qubit gates

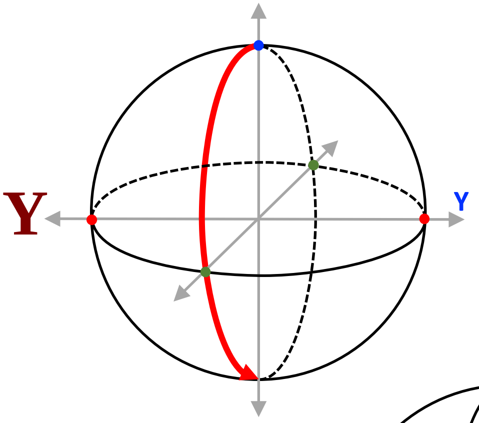
X

Pauli-X



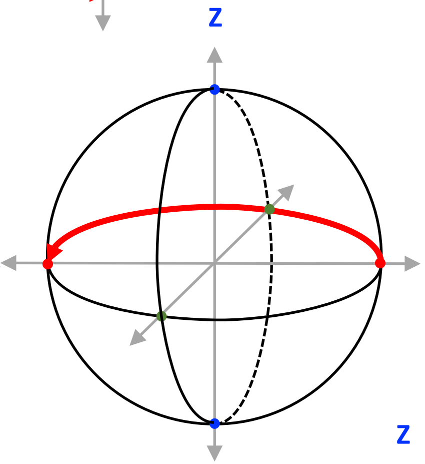
Y

Pauli-Y



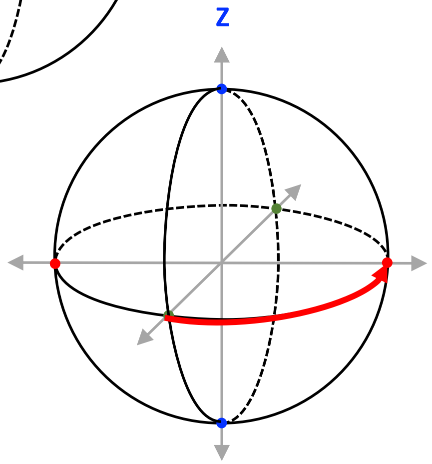
Z

Pauli-Z



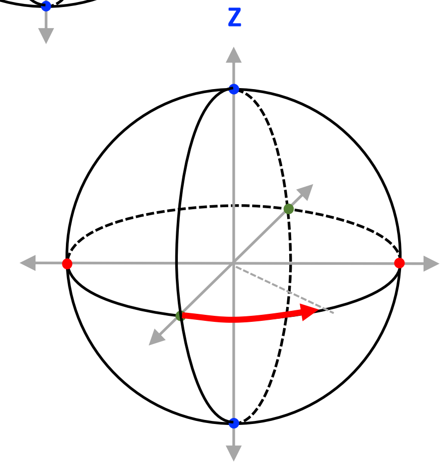
S

Phase (S, P)



T

π/8 (T)



the S-gate and T-gate are **not** their own inverse

Hadamard gate  to turn ON and OFF Superposition

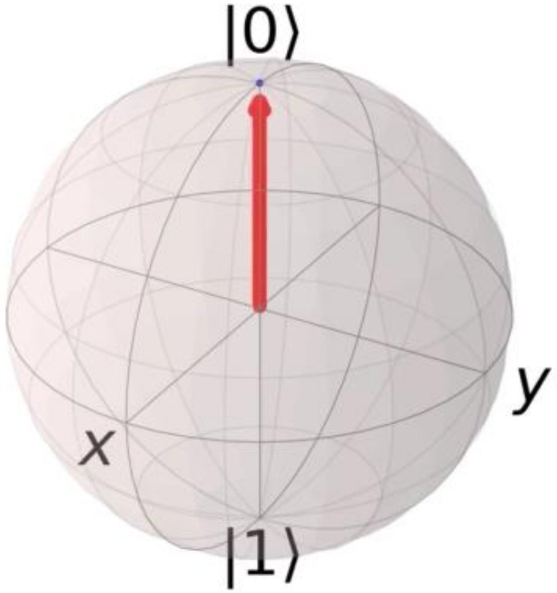
$H|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + 1\rangle) = |+\rangle$

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

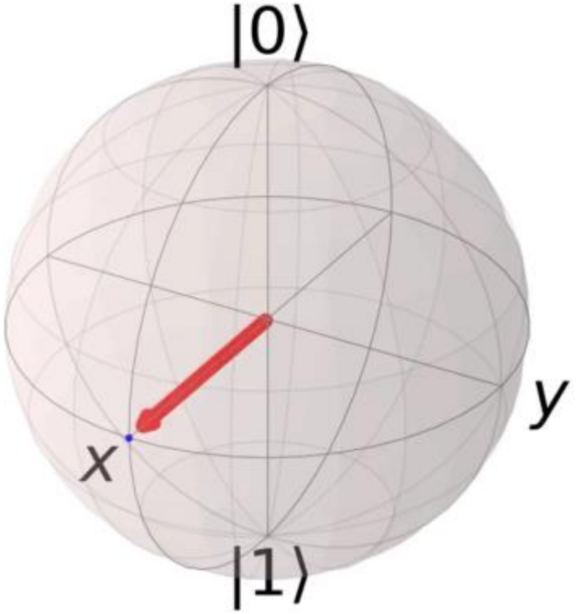
$H|1\rangle = \frac{1}{\sqrt{2}}(|0\rangle - 1\rangle) = |-\rangle$

$H|-\rangle = |1\rangle$

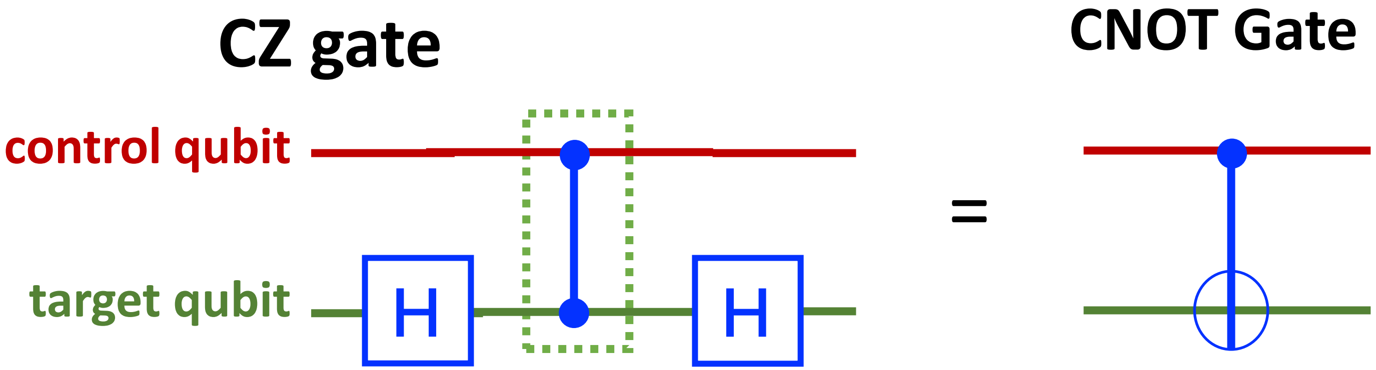
From $|0\rangle$ to superposition state



From superposition state back to $|0\rangle$



A basic conditional gate: Control Z gate

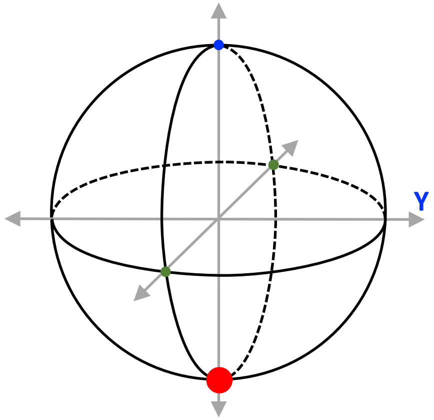


CZ gate

CNOT Gate

control qubit

target qubit



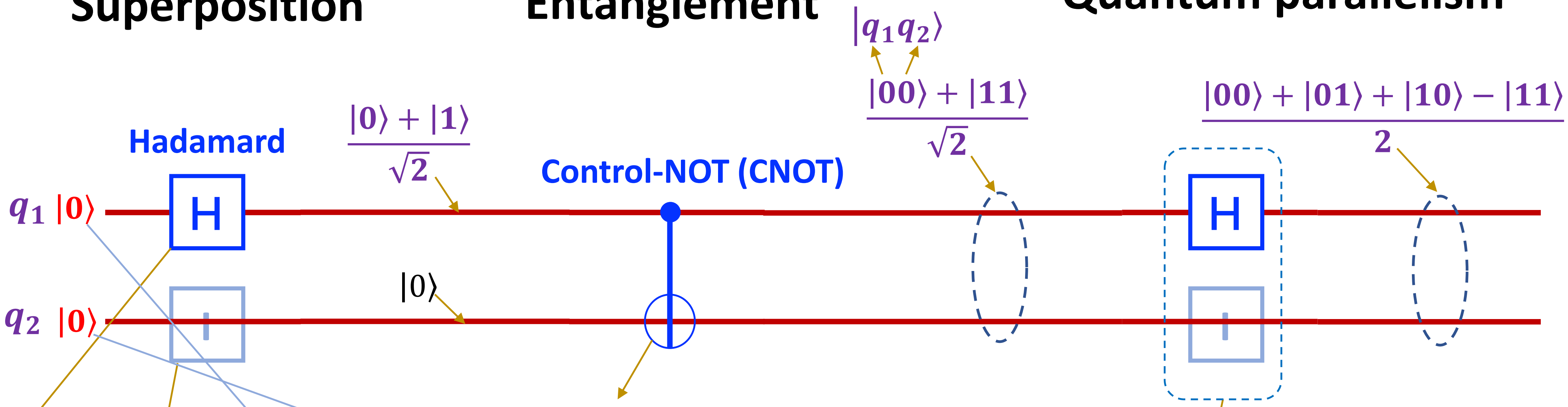
Control qubit = $|1\rangle$

Unveiling the Power of Quantum Logic Gates

Superposition

Entanglement

Quantum parallelism



$$\underbrace{\left(\frac{1}{\sqrt{2}}\begin{bmatrix}1&1\\1&-1\end{bmatrix}\otimes\begin{bmatrix}1&0\\0&1\end{bmatrix}\right)}\underbrace{\left(\begin{bmatrix}1\\0\end{bmatrix}\otimes\begin{bmatrix}1\\0\end{bmatrix}\right)}$$

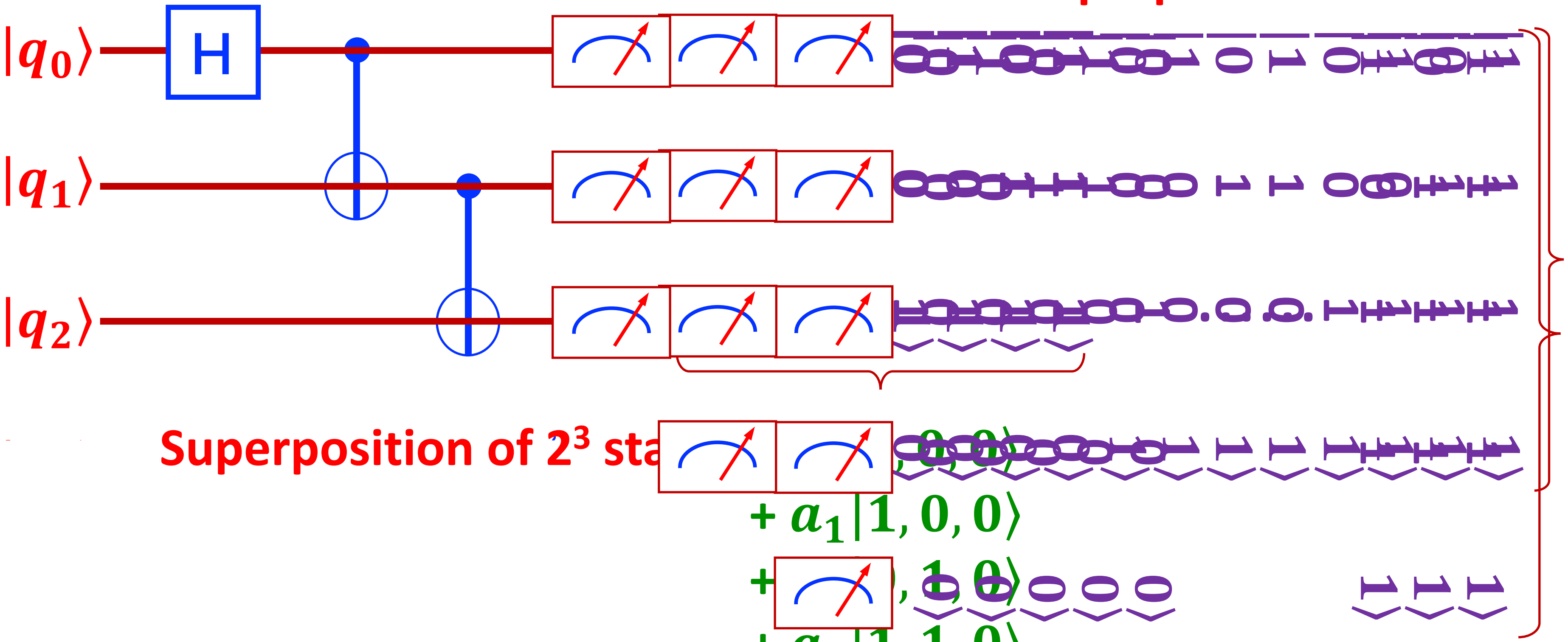
$$\frac{1}{\sqrt{2}}\begin{bmatrix}1&0&1&0\\0&1&0&1\\1&0&-1&0\\0&1&0&-1\end{bmatrix}\begin{pmatrix}1\\0\\0\\0\end{pmatrix}=\frac{1}{\sqrt{2}}\begin{pmatrix}1\\0\\1\\0\end{pmatrix}=\frac{|00\rangle+|10\rangle}{\sqrt{2}}$$

$$\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&0&1\\0&0&1&0\end{bmatrix}\frac{1}{\sqrt{2}}\begin{pmatrix}1\\0\\1\\0\end{pmatrix}=\frac{1}{\sqrt{2}}\begin{pmatrix}1\\0\\0\\1\end{pmatrix}=\frac{|00\rangle+|11\rangle}{\sqrt{2}}$$

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

$$=\frac{|00\rangle+|01\rangle+|10\rangle-|11\rangle}{2}$$

Harnessing the Power of Quantum Computers : Processing Superpositions of 2^n States



Superposition of 2^3 states :

$+ a_1|1,0,0\rangle$
$+ a_2|0,1,0\rangle$
$+ a_3|1,1,0\rangle$
$+ a_4|0,0,1\rangle$
$+ a_5|1,0,1\rangle$
$+ a_6|0,1,1\rangle$
$+ a_7|1,1,1\rangle$

Superposition of 2^4 states : $a_0|0,0,0,0\rangle$
$+ a_1|1,0,0,0\rangle$
$+ a_2|0,1,0,0\rangle$
$+ a_3|1,1,0,0\rangle$
$+ a_4|0,0,1,0\rangle$
$+ a_5|1,0,1,0\rangle$
$+ a_6|0,1,1,0\rangle$
$+ a_7|1,1,1,0\rangle$
$+ a_8|0,0,0,1\rangle$
$+ a_9|1,0,0,1\rangle$
$+ a_{10}|0,1,0,1\rangle$
$+ a_{11}|1,1,0,1\rangle$
$+ a_{12}|0,0,1,1\rangle$
$+ a_{13}|1,0,1,1\rangle$
$+ a_{14}|0,1,1,1\rangle$
$+ a_{15}|1,1,1,1\rangle$

Superposition of 2^5 states : $a_0|0,0,0,0,0\rangle$
$+ a_1|1,0,0,0,0\rangle$
$+ a_2|0,1,0,0,0\rangle$
$+ a_3|1,1,0,0,0\rangle$
$+ a_4|0,0,1,0,0\rangle$
$+ a_{29}|1,0,1,1,1\rangle$
$+ a_{30}|0,1,1,1,1\rangle$
$+ a_{31}|1,1,1,1,1\rangle$

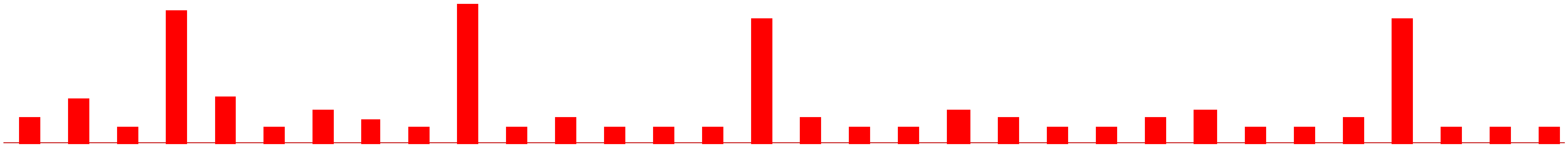
$2^{270} \approx 10^{81}$: number of atoms in the universe

(IBM Osprey QPU = 433 qubits)

Probability Distributions: an exploration of Quantum Mechanics principles

$$\begin{array}{c}\mathbf{q0}\\\mathbf{q1}\\\mathbf{q2}\\\mathbf{q3}\\\mathbf{q4}\end{array}\overbrace{\begin{bmatrix}0\\1\\1\\0\\1\end{bmatrix}\begin{bmatrix}1\\1\\1\\0\\0\end{bmatrix}\begin{bmatrix}1\\1\\0\\0\\0\end{bmatrix}\begin{bmatrix}0\\1\\0\\1\\0\end{bmatrix}\begin{bmatrix}1\\0\\0\\0\\1\end{bmatrix}\begin{bmatrix}0\\0\\1\\1\\0\end{bmatrix}\begin{bmatrix}0\\1\\1\\0\\1\end{bmatrix}\begin{bmatrix}1\\0\\0\\0\\1\end{bmatrix}\begin{bmatrix}0\\1\\0\\1\\0\end{bmatrix}\begin{bmatrix}1\\1\\1\\0\\0\end{bmatrix}\begin{bmatrix}1\\1\\1\\0\\0\end{bmatrix}\begin{bmatrix}0\\1\\0\\1\\0\end{bmatrix}\begin{bmatrix}1\\0\\0\\0\\1\end{bmatrix}\begin{bmatrix}0\\1\\1\\0\\1\end{bmatrix}\begin{bmatrix}0\\0\\1\\1\\0\end{bmatrix}\cdots\cdots\cdots\begin{bmatrix}1\\0\\0\\1\\0\end{bmatrix}\begin{bmatrix}0\\0\\1\\1\\0\end{bmatrix}\begin{bmatrix}0\\0\\1\\1\\0\end{bmatrix}\begin{bmatrix}1\\1\\1\\0\\0\end{bmatrix}\begin{bmatrix}0\\0\\1\\1\\0\end{bmatrix}}^{\textbf{10,000 measurements}}$$

Possibility distribution of 2^5 states



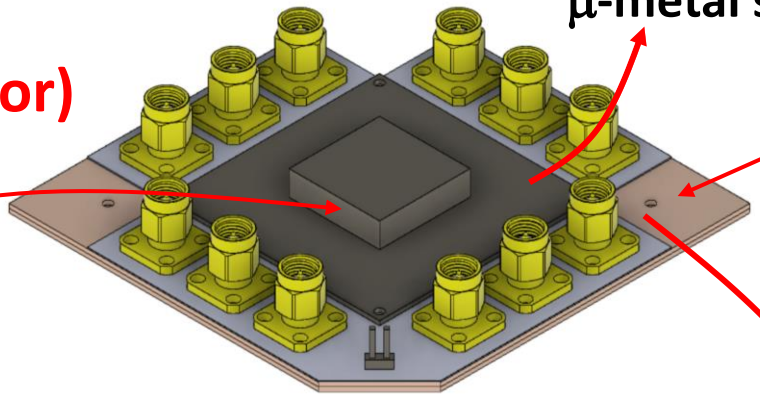
$$\begin{array}{c}\mathbf{q0}\\\mathbf{q1}\\\mathbf{q2}\\\mathbf{q3}\\\mathbf{q4}\end{array}\begin{bmatrix}0\\0\\0\\0\\0\end{bmatrix}\begin{bmatrix}1\\0\\0\\0\\0\end{bmatrix}\begin{bmatrix}0\\1\\0\\0\\0\end{bmatrix}\begin{bmatrix}1\\1\\0\\0\\0\end{bmatrix}\begin{bmatrix}0\\0\\1\\0\\0\end{bmatrix}\begin{bmatrix}1\\0\\1\\0\\0\end{bmatrix}\begin{bmatrix}0\\1\\1\\0\\0\end{bmatrix}\begin{bmatrix}1\\1\\1\\0\\0\end{bmatrix}\begin{bmatrix}0\\0\\0\\1\\0\end{bmatrix}\begin{bmatrix}1\\0\\0\\1\\0\end{bmatrix}\begin{bmatrix}0\\1\\0\\1\\0\end{bmatrix}\begin{bmatrix}1\\1\\0\\1\\0\end{bmatrix}\begin{bmatrix}0\\0\\1\\1\\0\end{bmatrix}\begin{bmatrix}1\\0\\1\\1\\0\end{bmatrix}\begin{bmatrix}0\\1\\1\\1\\0\end{bmatrix}\begin{bmatrix}1\\1\\1\\1\\0\end{bmatrix}\begin{bmatrix}0\\0\\0\\0\\1\end{bmatrix}\begin{bmatrix}1\\0\\0\\0\\1\end{bmatrix}\begin{bmatrix}0\\1\\0\\0\\1\end{bmatrix}\begin{bmatrix}1\\1\\0\\0\\1\end{bmatrix}\begin{bmatrix}0\\0\\1\\0\\1\end{bmatrix}\begin{bmatrix}1\\0\\1\\0\\1\end{bmatrix}\begin{bmatrix}0\\1\\1\\0\\1\end{bmatrix}\begin{bmatrix}1\\1\\1\\0\\1\end{bmatrix}\begin{bmatrix}0\\0\\0\\1\\1\end{bmatrix}\begin{bmatrix}1\\0\\0\\1\\1\end{bmatrix}\begin{bmatrix}0\\1\\0\\1\\1\end{bmatrix}\begin{bmatrix}1\\1\\0\\1\\1\end{bmatrix}\begin{bmatrix}0\\0\\1\\1\\1\end{bmatrix}\begin{bmatrix}1\\0\\1\\1\\1\end{bmatrix}\begin{bmatrix}0\\1\\1\\1\\1\end{bmatrix}\begin{bmatrix}1\\1\\1\\1\\1\end{bmatrix}$$

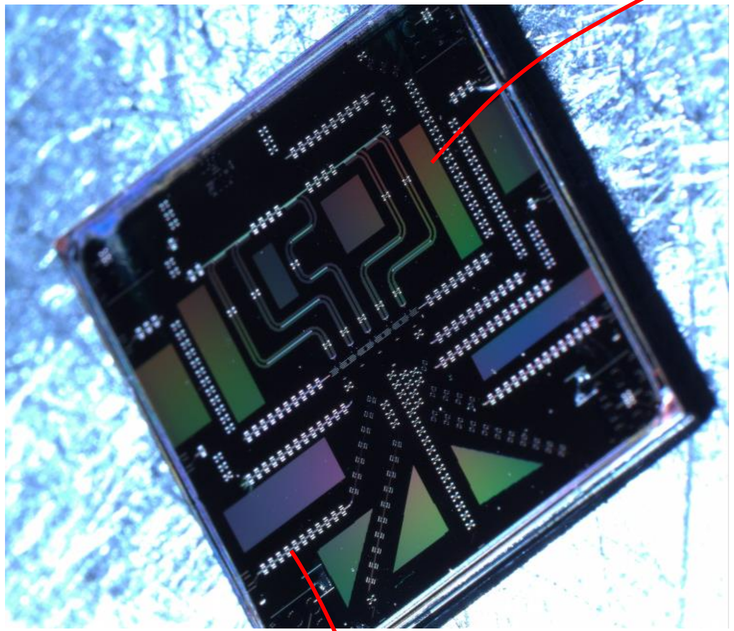
Building a Superconducting Quantum Computing :

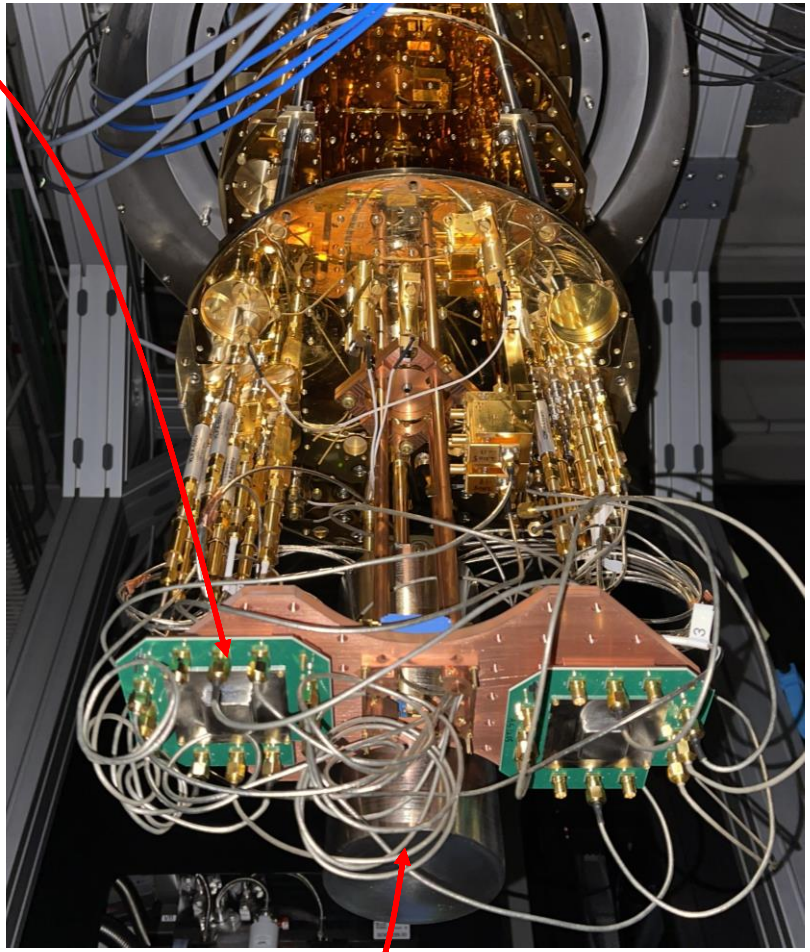
Chip Design
Device Fabrication
Packaging and Measurement Circuits
Readout and Control signals
Software Stack

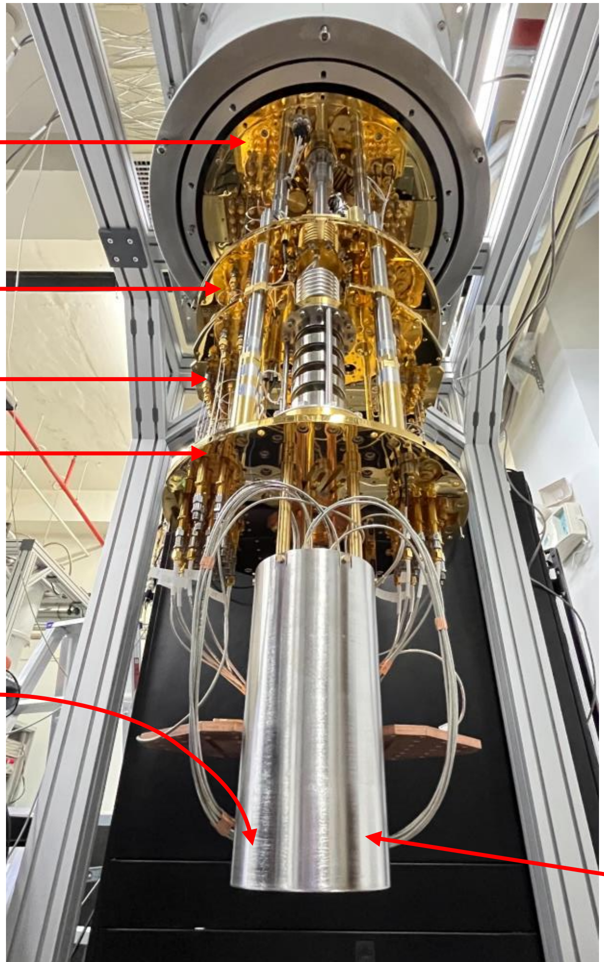
QPU Packaging

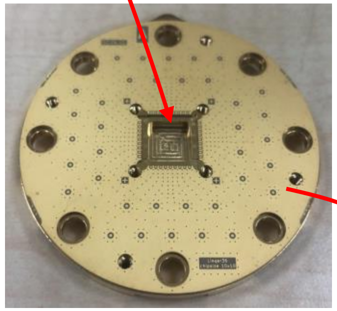
(bonding, shielding, thermal anchor)

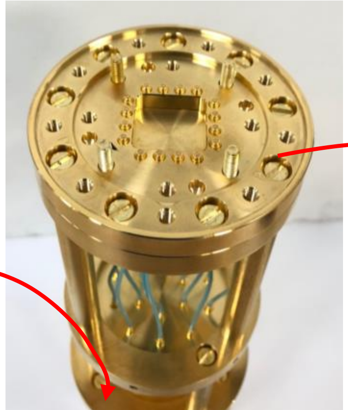








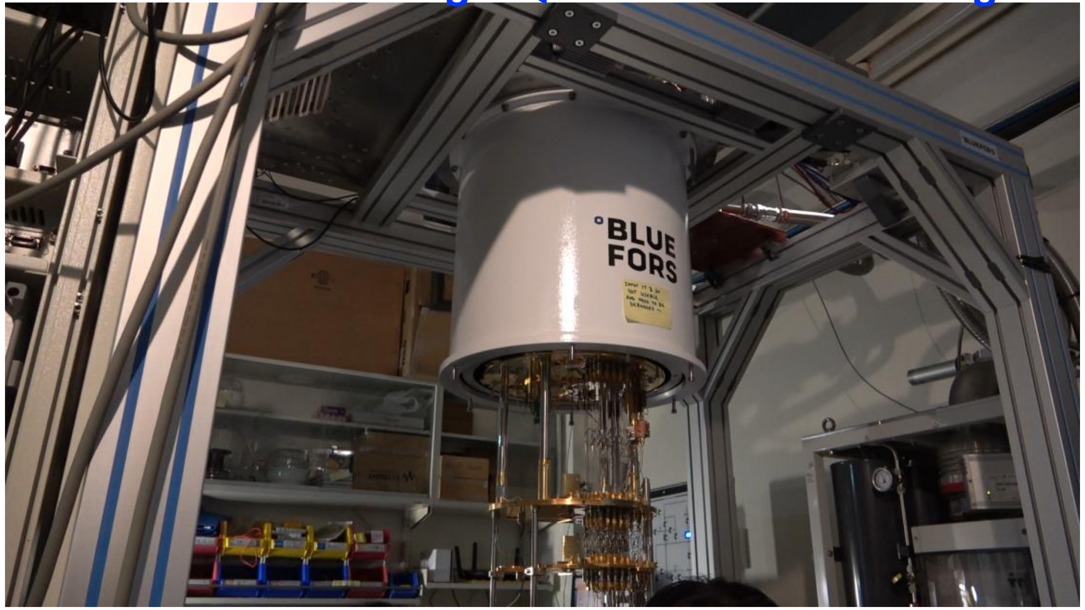




QPU fabrication：electron beam exposure

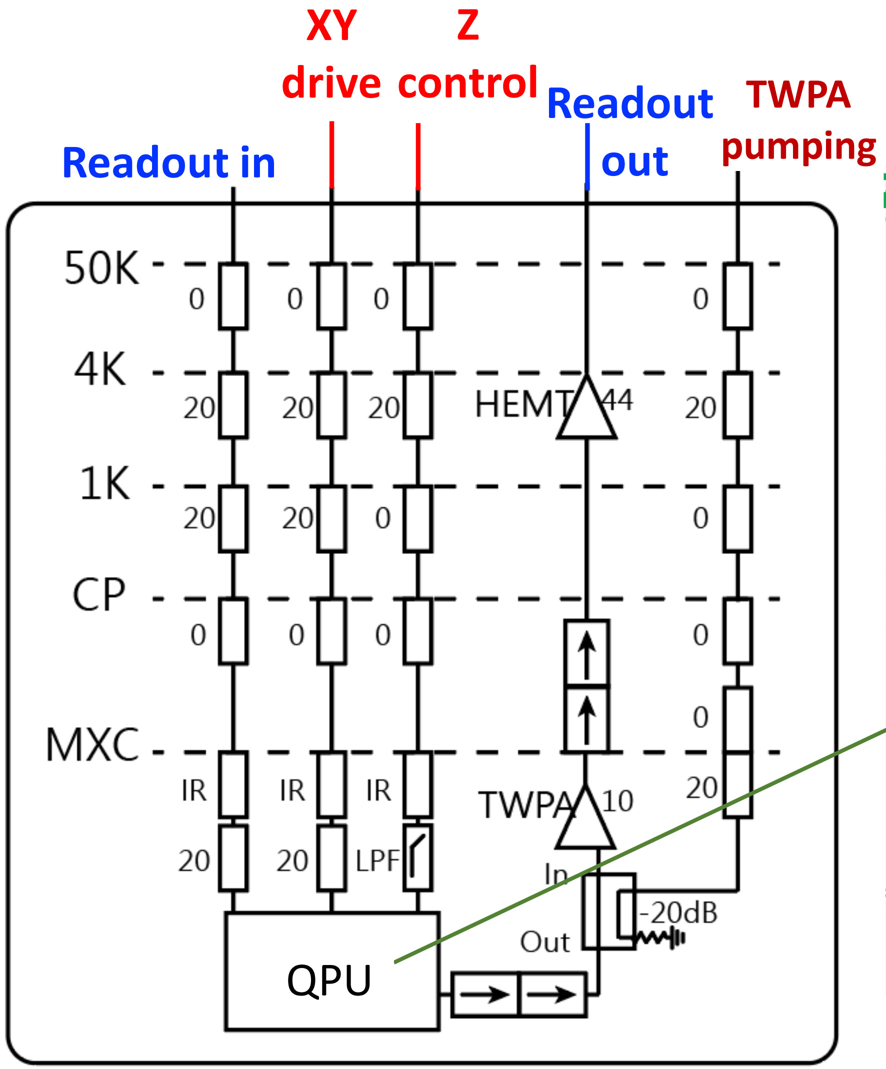
QPU fabrication：surface cleaning/thin film deposition

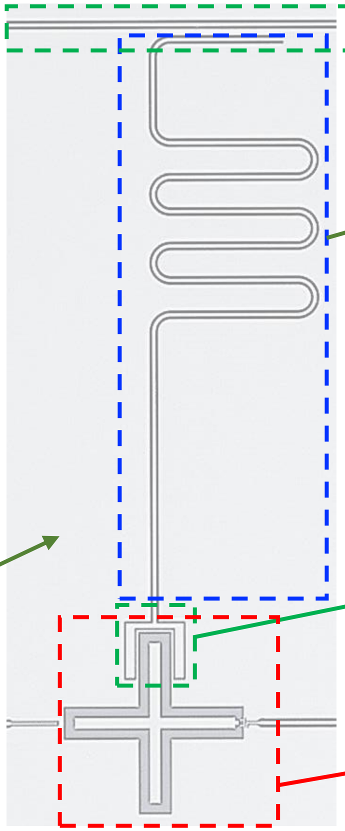
Measurement：loading a QPU into a dilution fridge



Measurement：control and readout of qubit states

Measurement Circuit and QPU parameters





Resonator-feedline coupling Qc: 7000 (kc: $1MHz$)

Resonator:

frequency: $5.7 \sim 6.3\ GHz$

Q_i : $> 10^5$ ($\sim 10^2$ photon number)
$> 10^6$ ($\sim 10^6$ photon number)

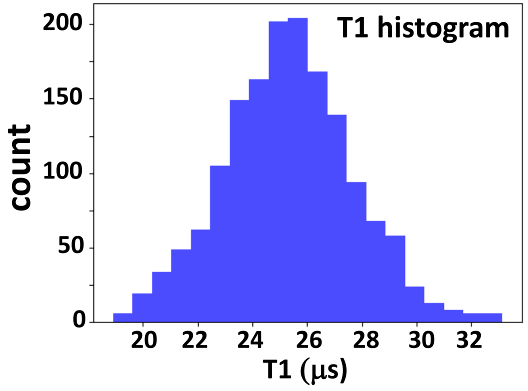
Qubit-Resonator coupling g : 60 MHz (~17ns)

Qubit:

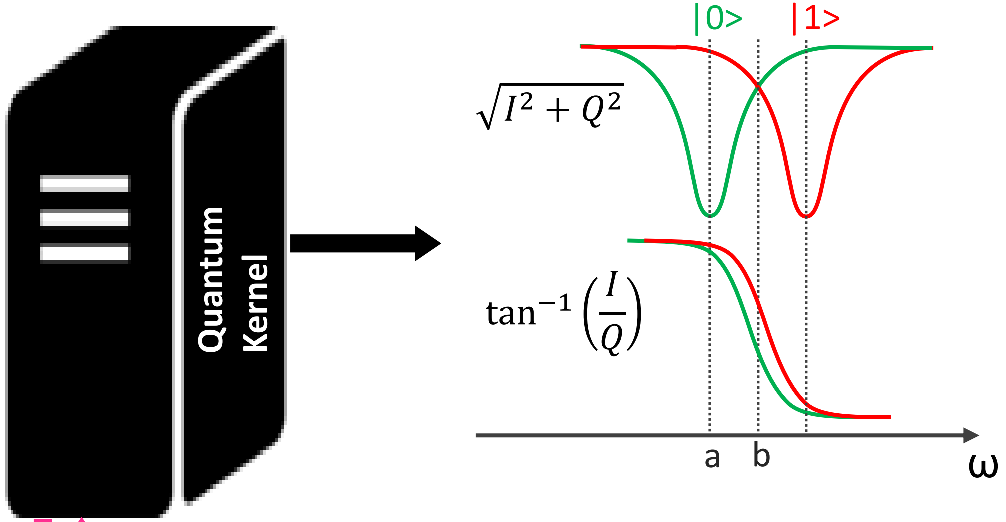
frequency: $3 \sim 5$ GHz

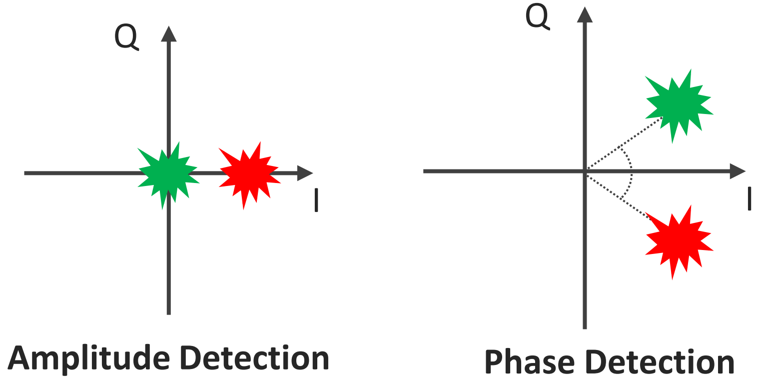
T1: $20 \sim 30\ \mu s$

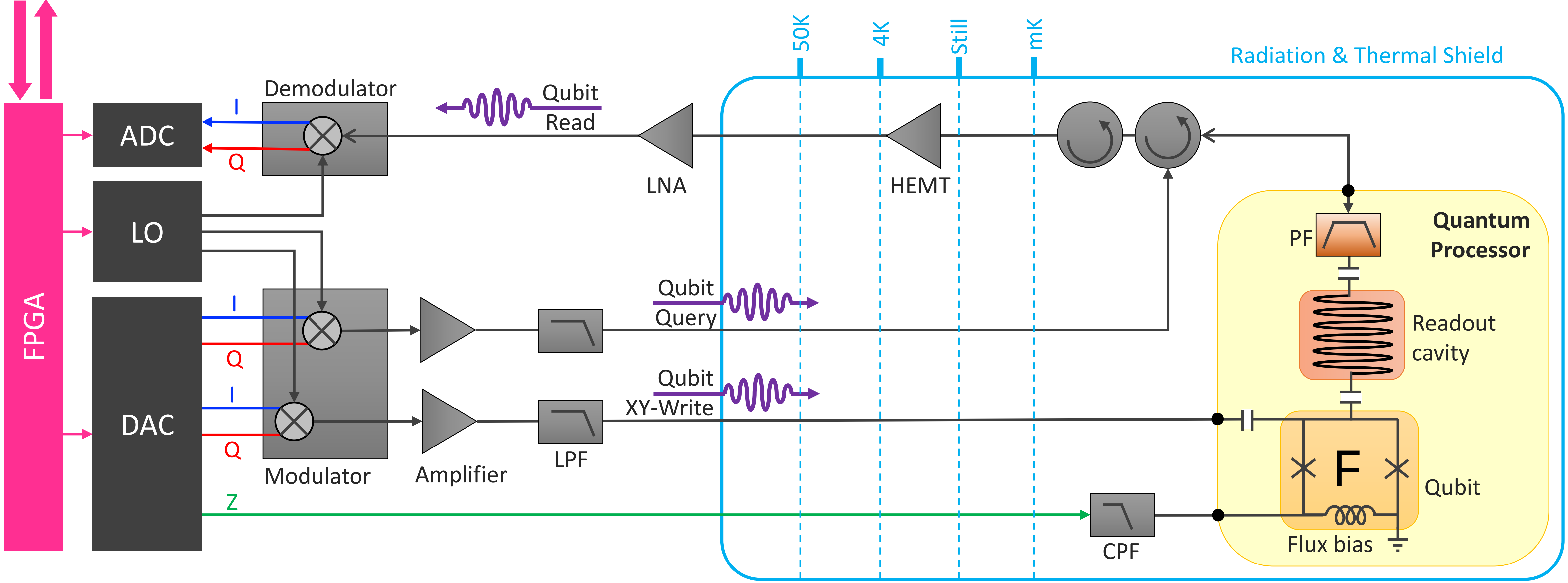
T2: $2 \sim 20\ \mu s$



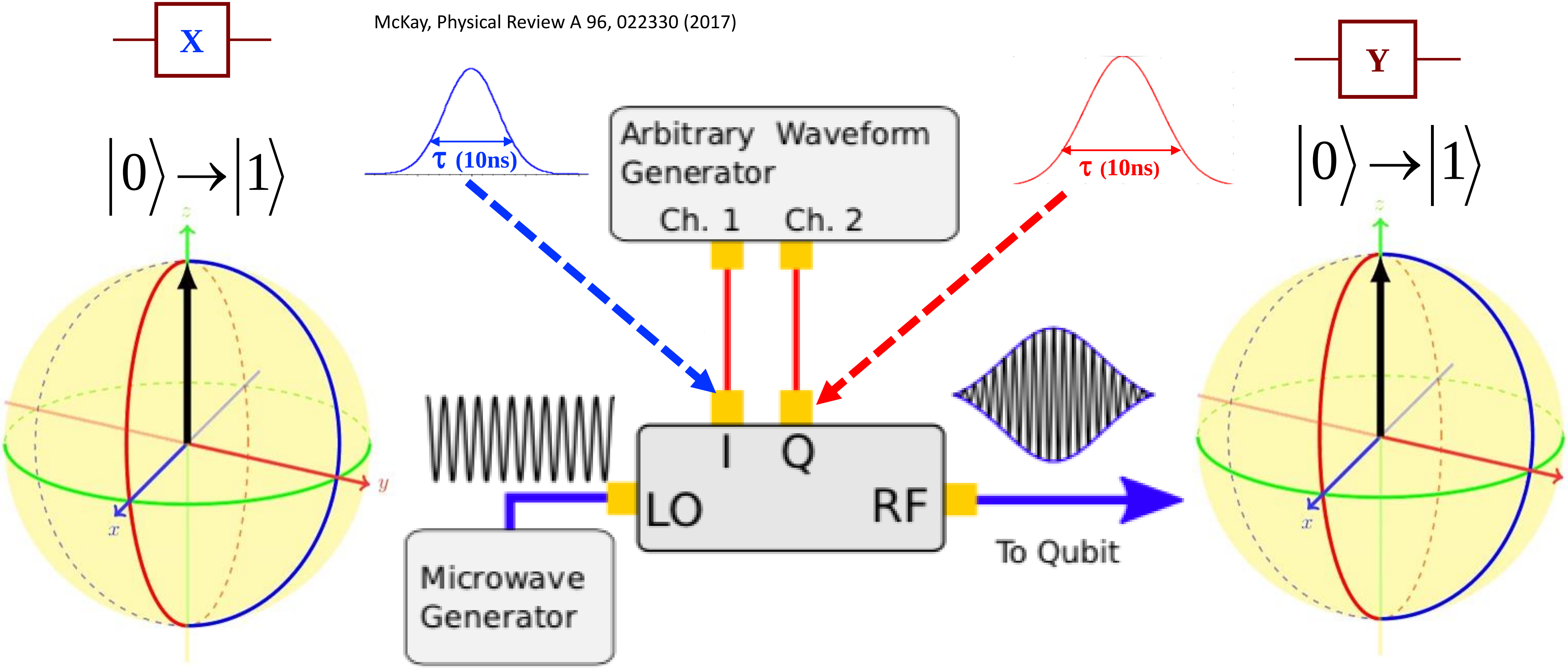
Complete Setup for Single Qubit Operation:
Vector XY-Control
Full Readout







X, Y-gates : rotating 180° with respect to X, Y-axis



McKay, Physical Review A 96, 022330 (2017)

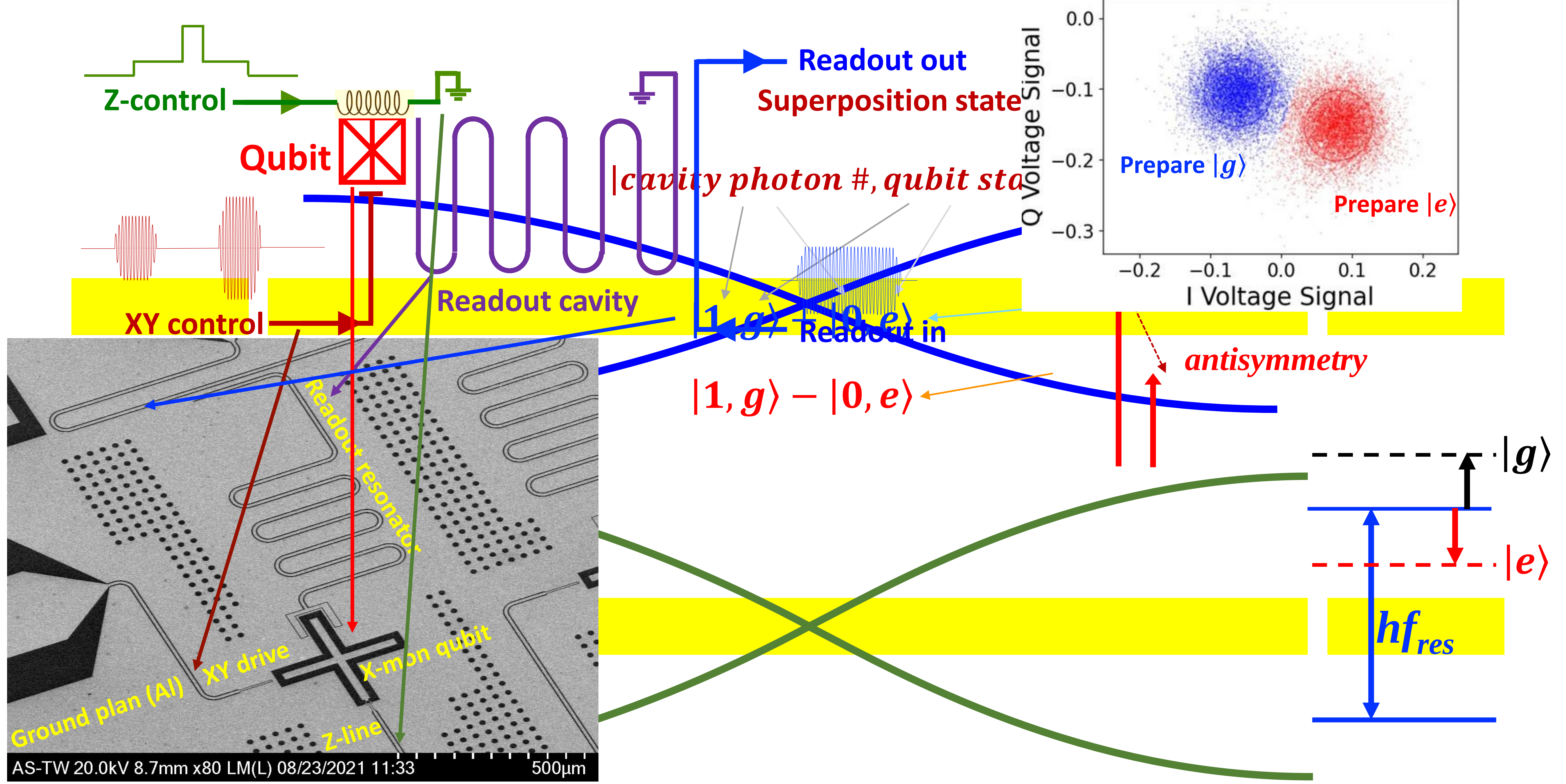
$|0\rangle \rightarrow |1\rangle$

$|0\rangle \rightarrow |1\rangle$

Readout of qubit states :

Entanglement between qubit and cavity photons

qubit



Z-control

Qubit

XY control

Readout cavity

Readout out

Readout in

Superposition state

$|cavity\ photon\ \#, qubit\ sta$

$|1, g\rangle + |0, e\rangle$

$|1, g\rangle - |0, e\rangle$

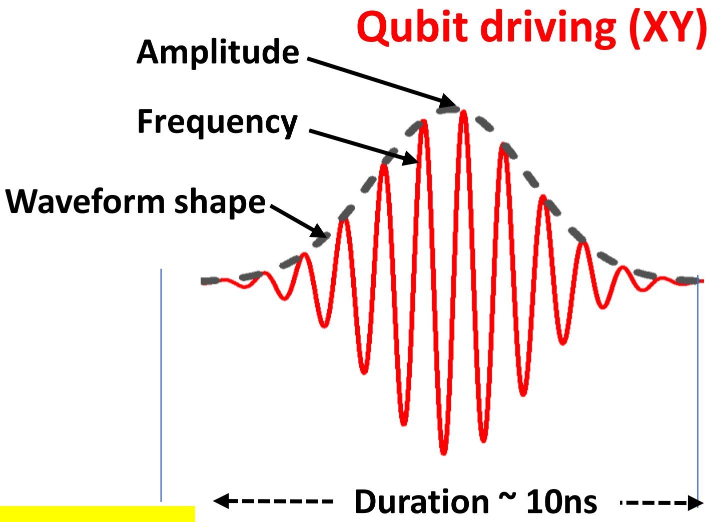
antisymmetry

$|g\rangle$

$|e\rangle$

hf_{res}

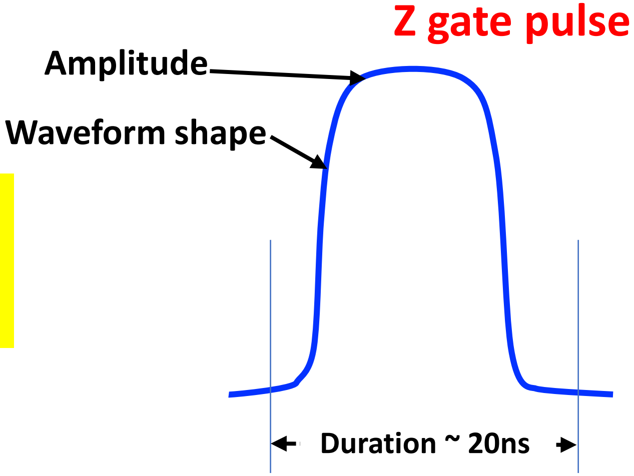
Qubit state control and readout



For single-qubit operation:

1. Optimizing Readout Frequency and Readout Power
2. Optimizing Driving (XY) Frequency and Power, fine tuning the Driving Waveform
3. **Auto-calibration: Routine Optimization of Driving Signals for Sustaining High Gate Fidelity**

For two-qubit operation:

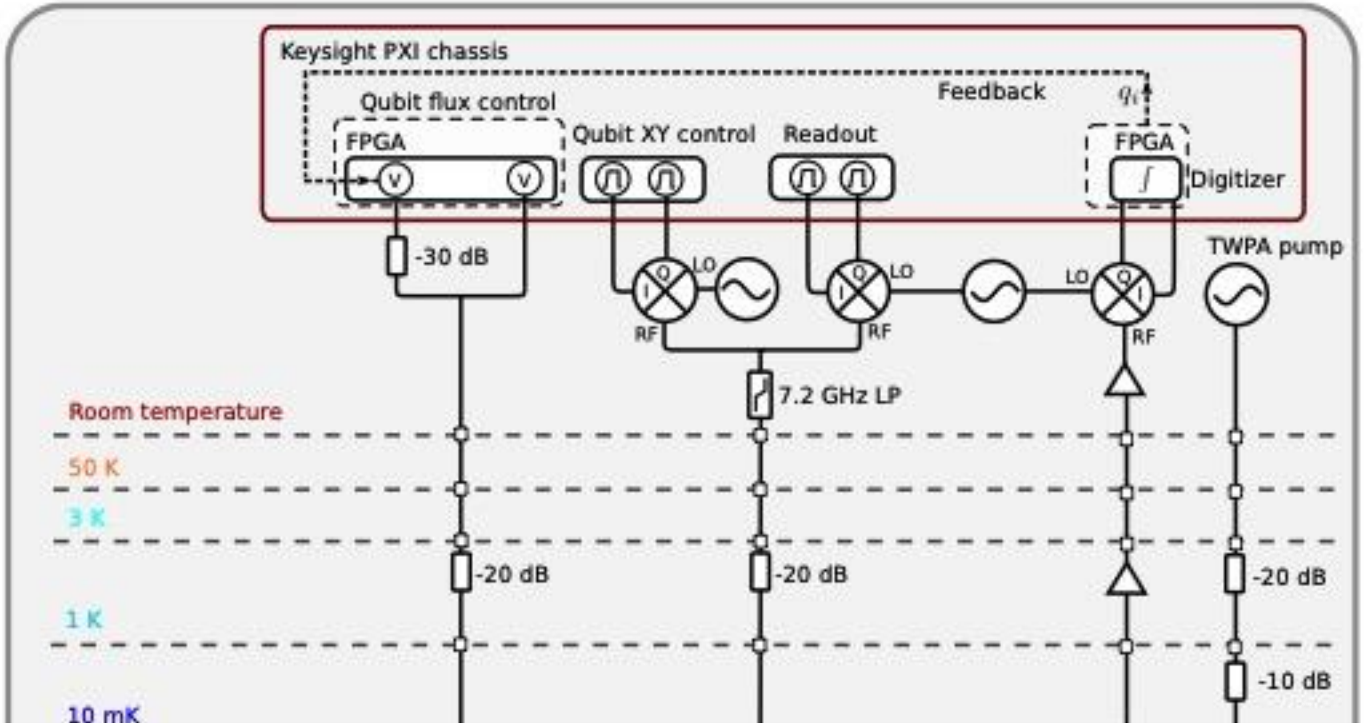


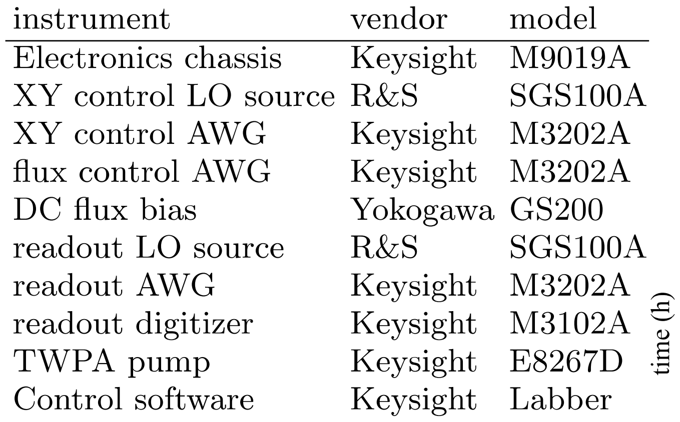
1. Optimizing frequency control (Z) pulse, fine tuning the Driving Waveform
2. **Auto-calibration: Routine Optimization of Z-pulse shapes for Sustaining High 2Q Gate Fidelity**

closed-loop feedback

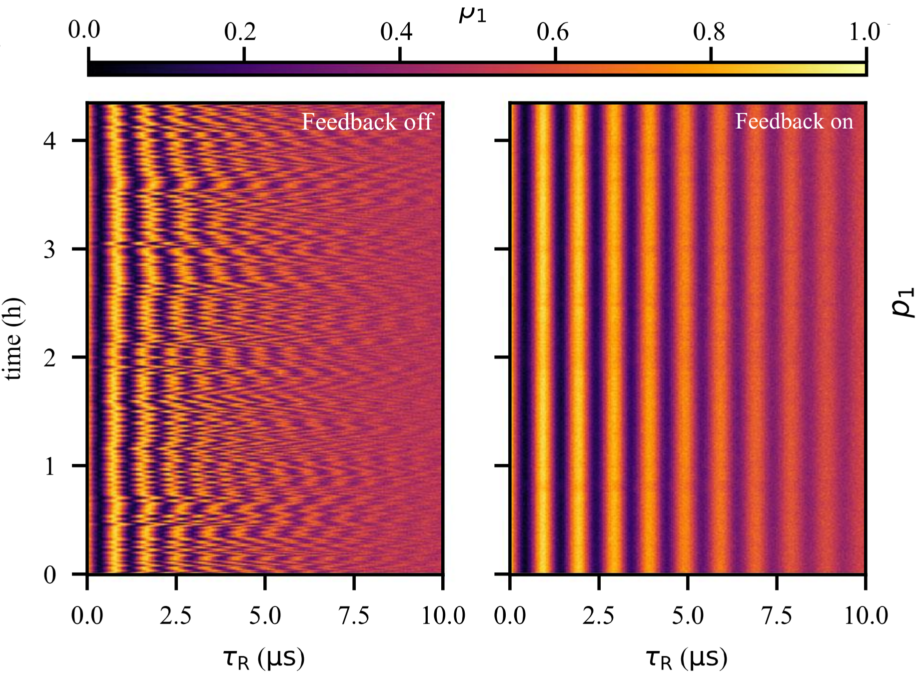
MIT, William D. Oliver group, MIT
Nature Comm. 2022
https://doi.org/10.1038/s41467-022-29287-4

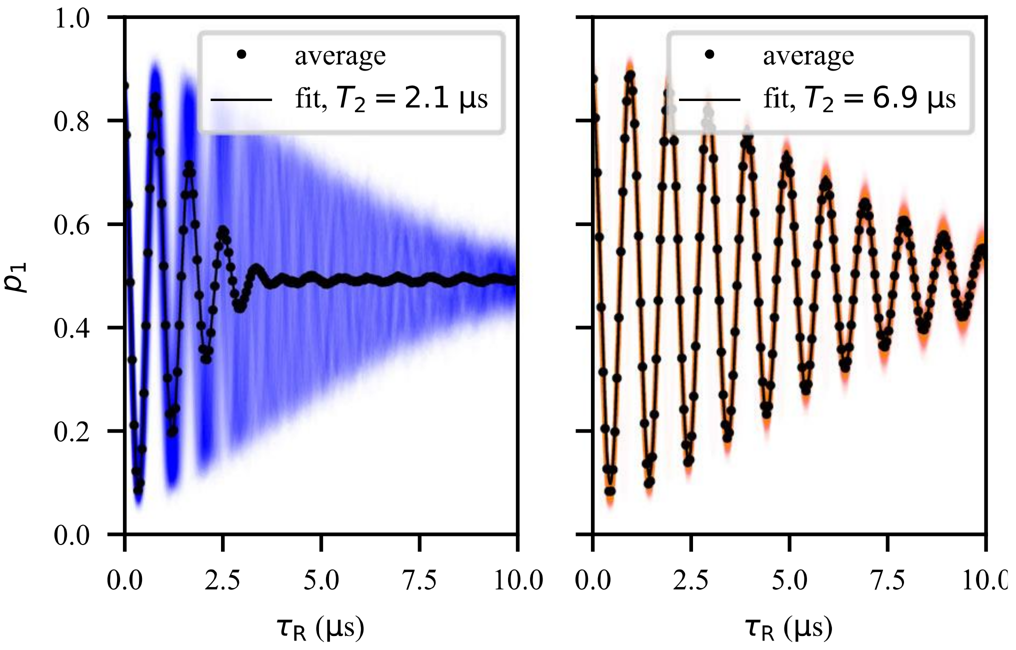
Note the feedback circuit



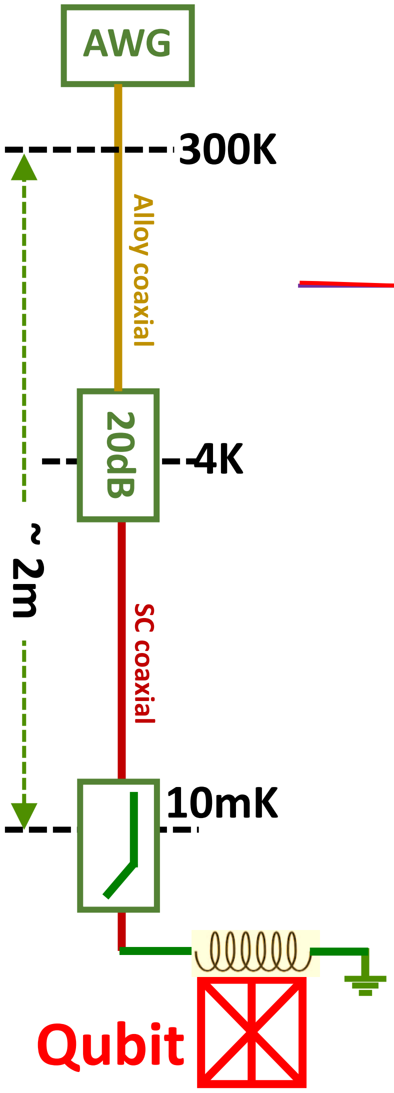


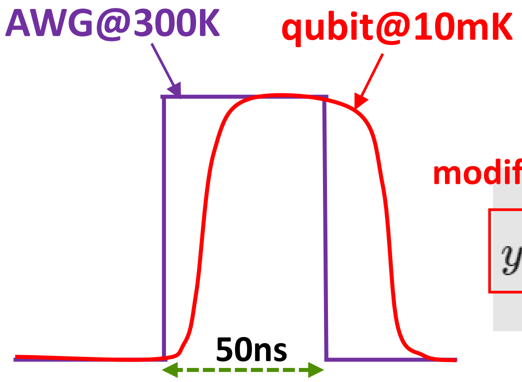
instrument	vendor	model
Electronics chassis	Keysight	M9019A
XY control LO source	R&S	SGS100A
XY control AWG	Keysight	M3202A
flux control AWG	Keysight	M3202A
DC flux bias	Yokogawa	GS200
readout LO source	R&S	SGS100A
readout AWG	Keysight	M3202A
readout digitizer	Keysight	M3102A
TWPA pump	Keysight	E8267D
Control software	Keysight	Labber





Digital Filter for Z-pulse shaping (cryoscope)



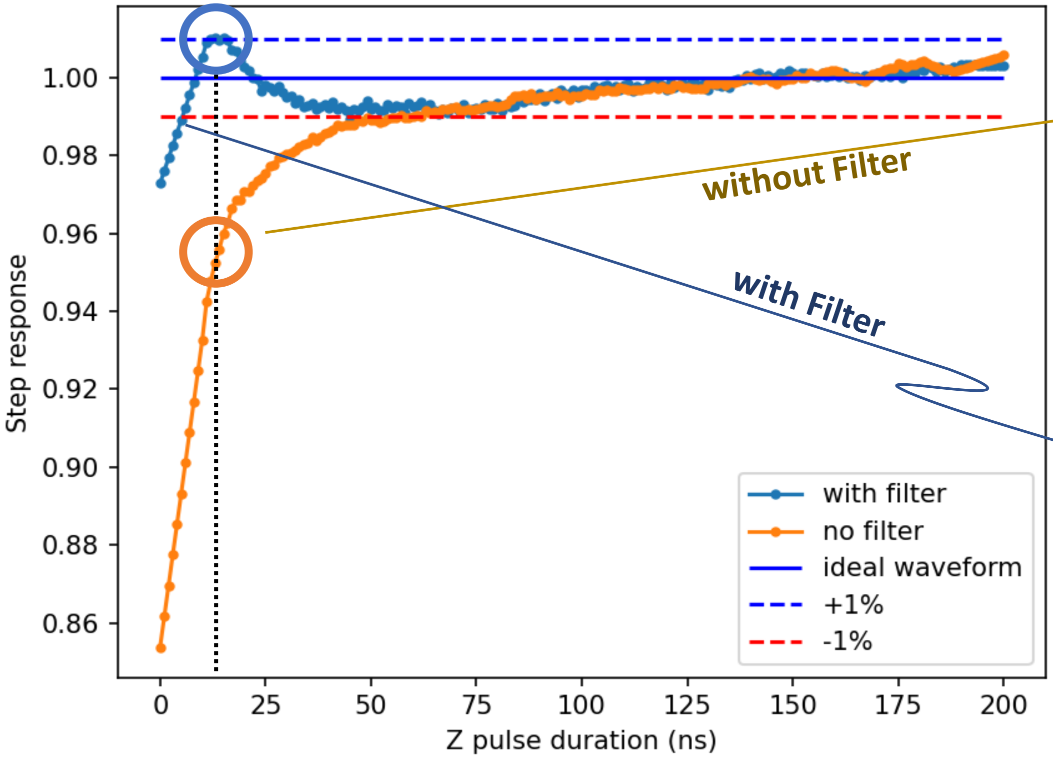


modified

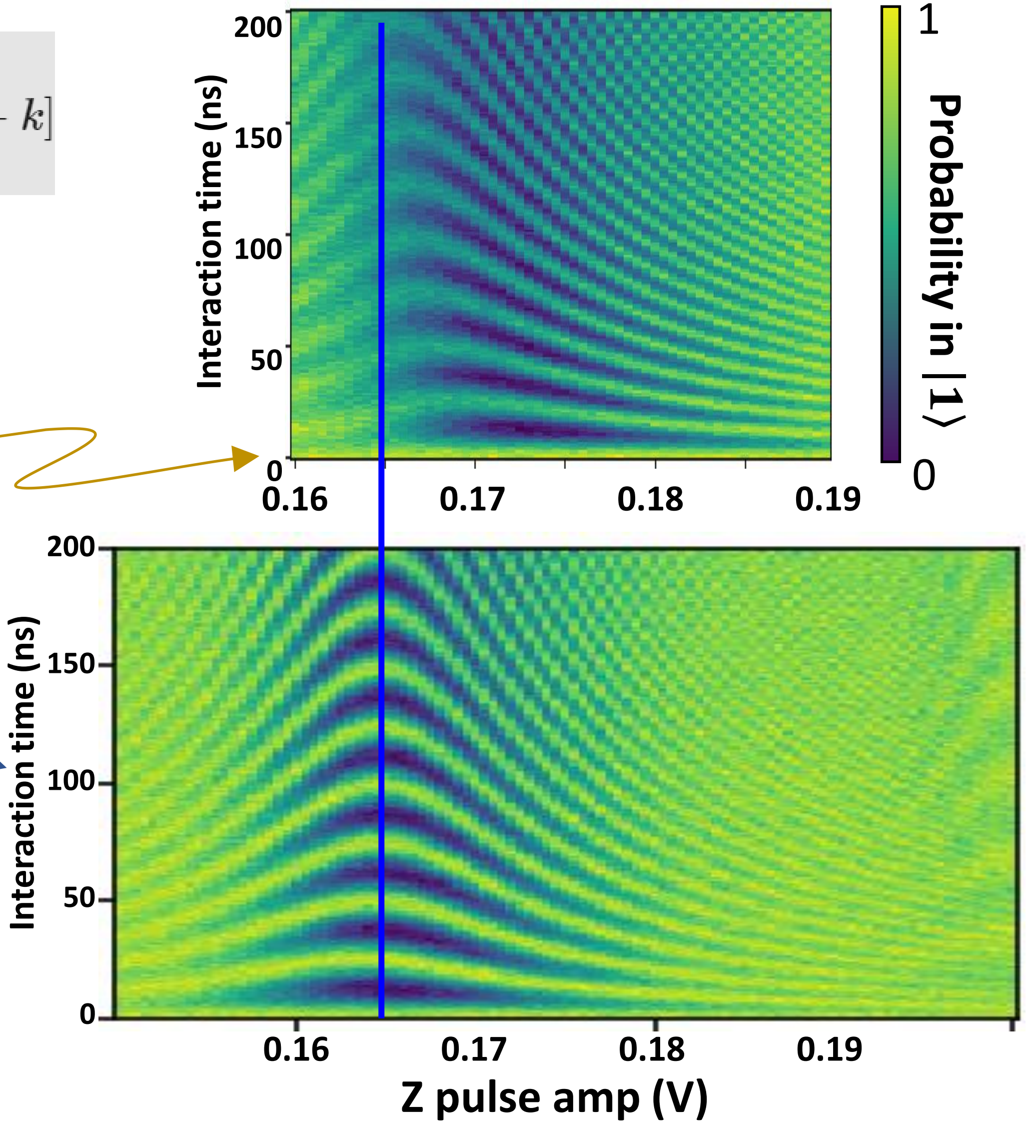
n : evolution

ideal

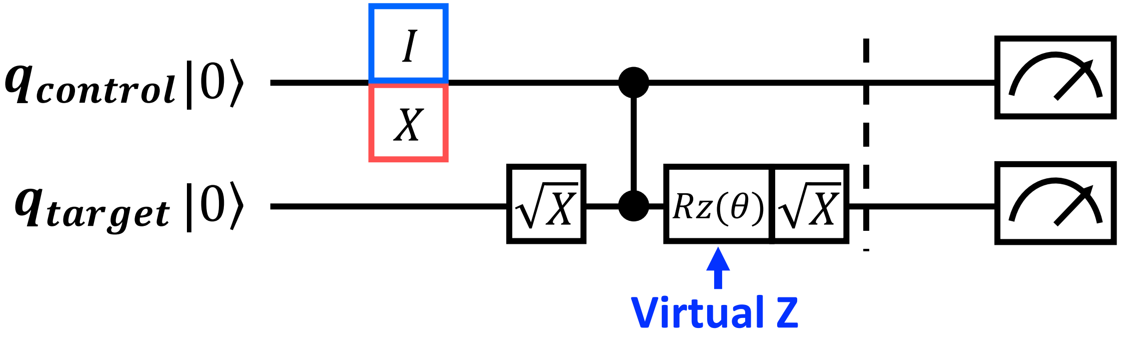
$$\boxed{y[n]} = \sum_{m=1}^{M} a_m y\boxed{[n} - m] + \sum_{k=0}^{K} b_k x\boxed{[n} - k]$$



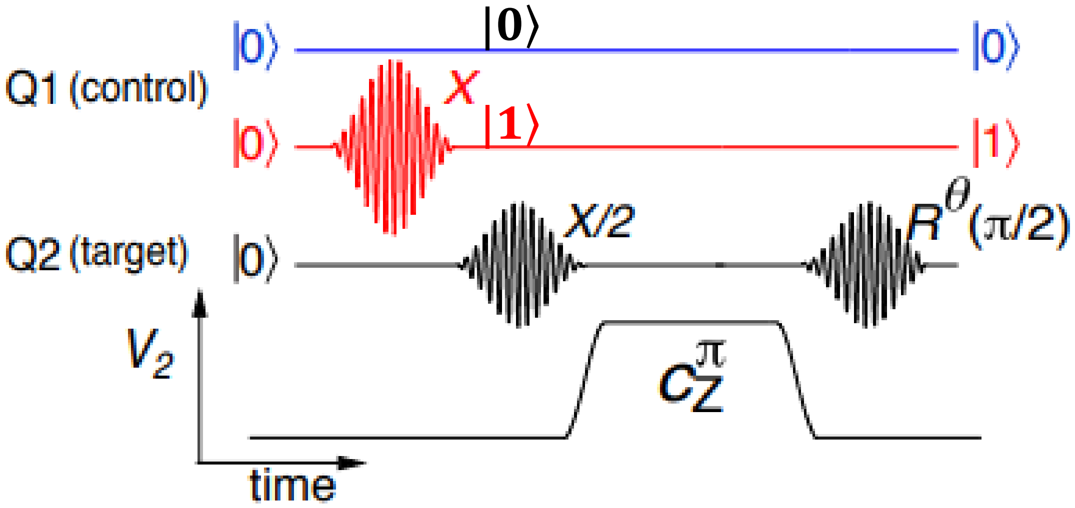
CZ chevron plot



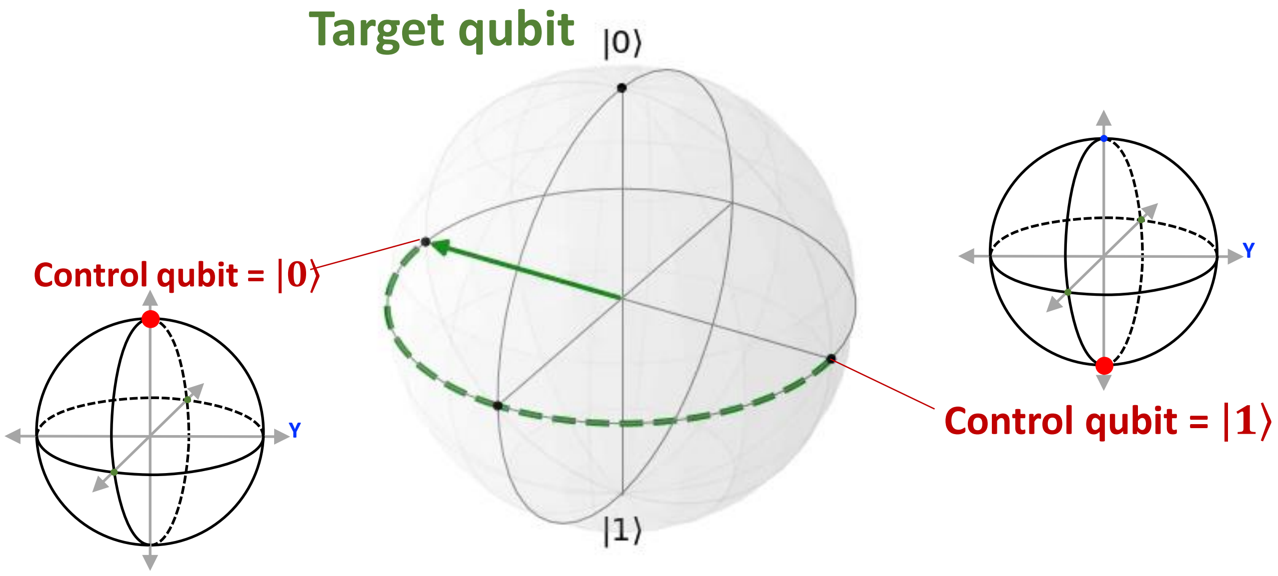
CZ Gate :



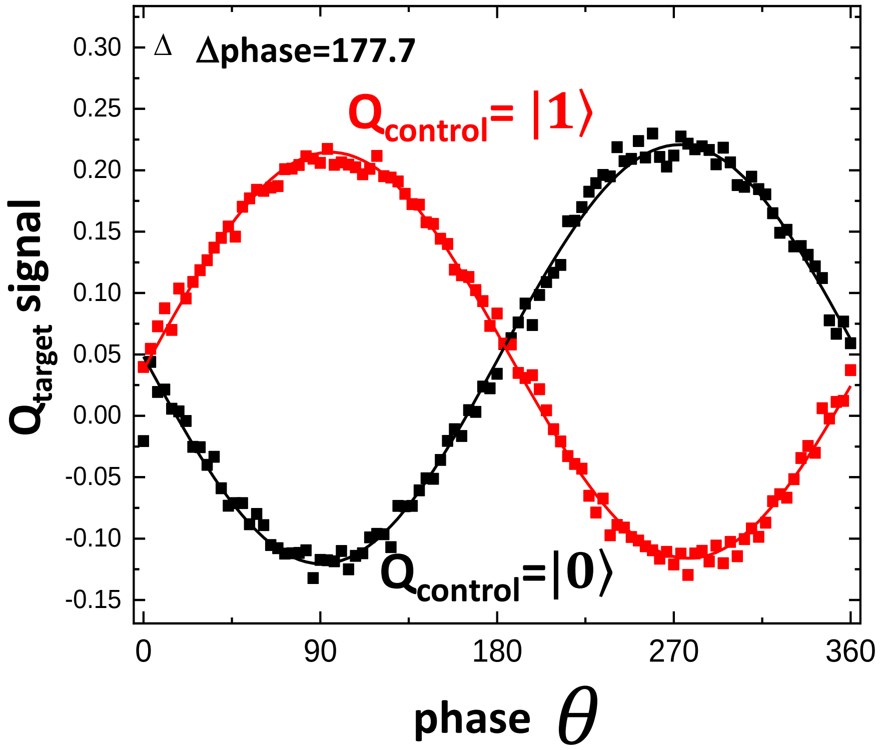
$q_{control}|0\rangle$
$q_{target}|0\rangle$
I
X
$\sqrt{X}$
$Rz(\theta)$
$\sqrt{X}$
Virtual Z



Q1 (control)
|0⟩
|0⟩
|0⟩
X
|1⟩
|0⟩
|1⟩
Q2 (target)
|0⟩
X/2
$R^{\theta}(\pi/2)$
V_2
C_Z^{π}
time



Target qubit
|0⟩
|1⟩
Control qubit = |0⟩
Control qubit = |1⟩
Y
Y

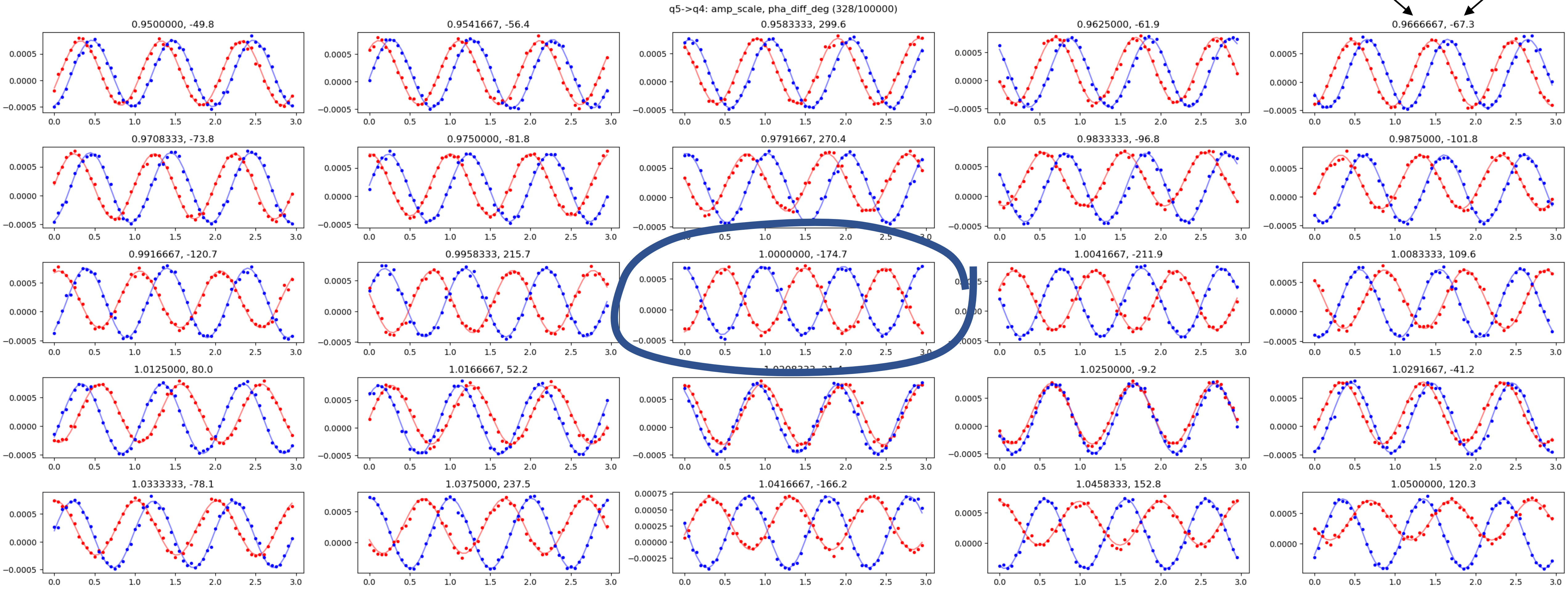


Δphase=177.7
$Q_{control}=|1\rangle$
$Q_{control}=|0\rangle$
Q_{target} signal
phase θ
0.30
0.25
0.20
0.15
0.10
0.05
0.00
-0.05
-0.10
-0.15
0
90
180
270
360

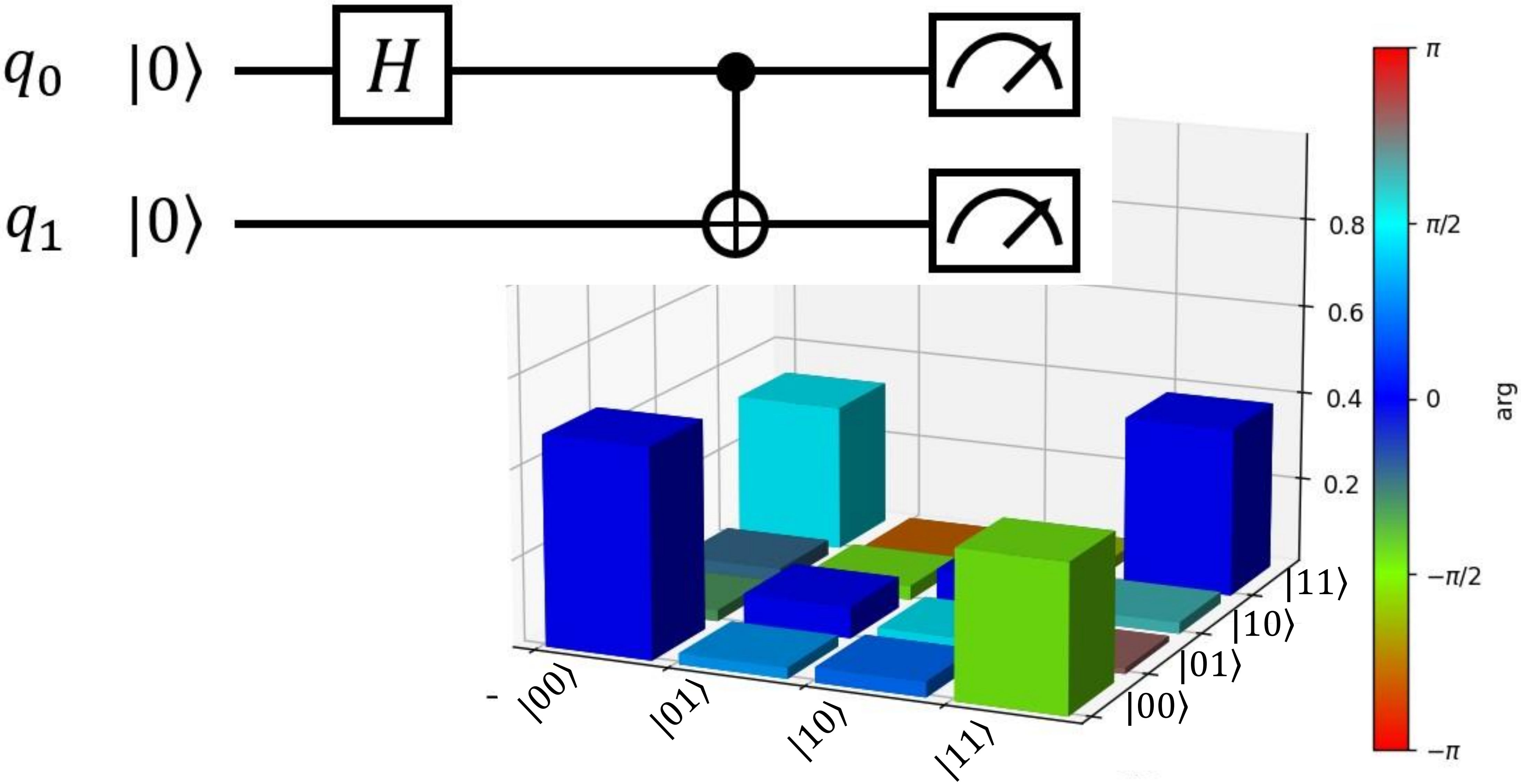
Searching for optimal **z-gate height**

Blue = Control $|0\rangle$ **Red = Control $|1\rangle$**

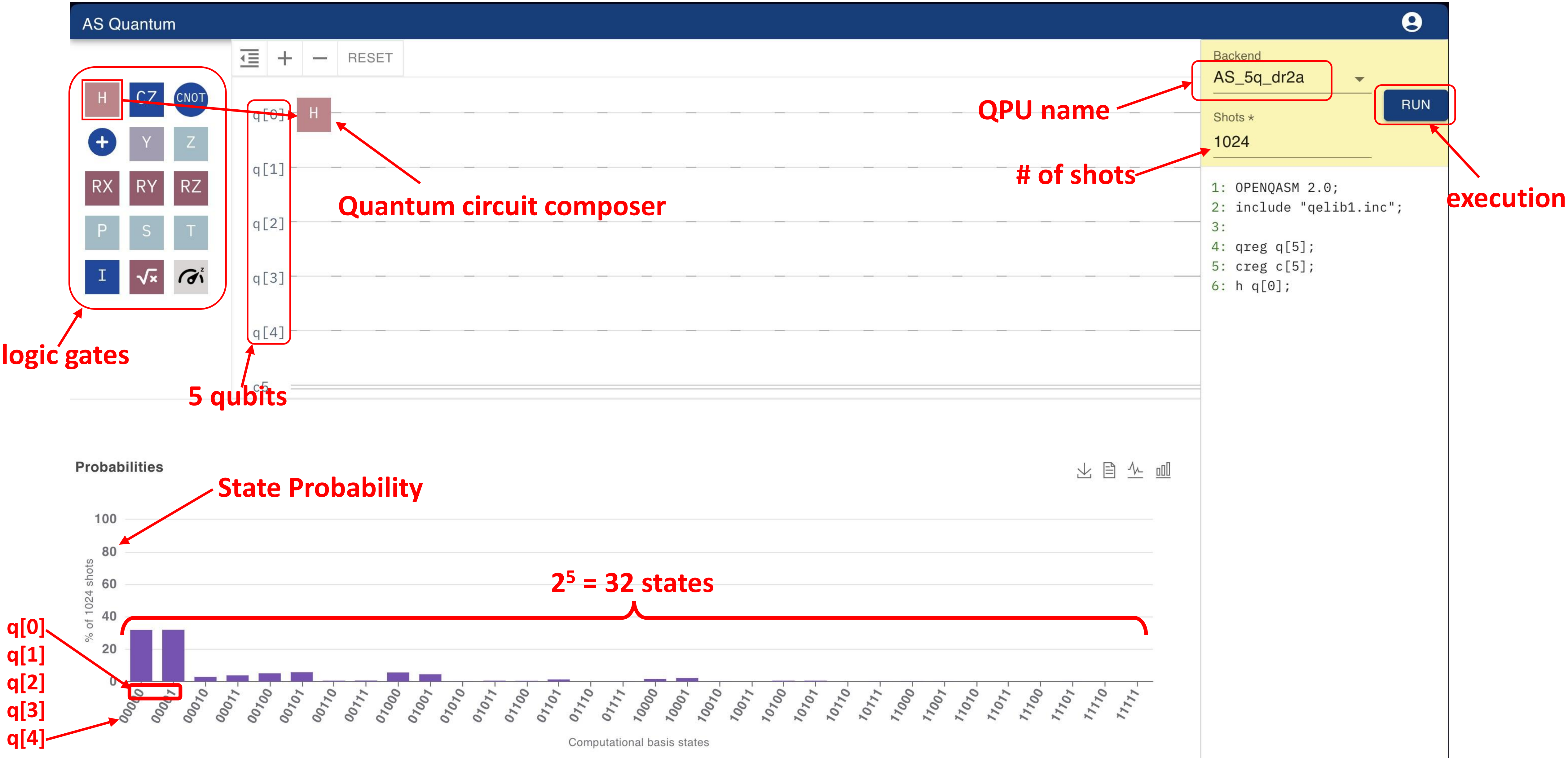
amplitude ratio, phase difference (deg)



BELL state


q_0 $|0\rangle$
H
q_1 $|0\rangle$
0.8
0.6
0.4
0.2
$|00\rangle$
$|01\rangle$
$|10\rangle$
$|11\rangle$
$|00\rangle$
$|01\rangle$
$|10\rangle$
$|11\rangle$
π
$\pi/2$
0
$-\pi/2$
$-\pi$
arg

USER interface



Backend: AS_5q_dr2a

Shots *: 1024

```
OPENQASM 2.0;
include "qelib1.inc";

qreg q[5];
creg c[5];
h q[0];
```

QPU name

# of shots

execution

Quantum circuit composer

logic gates

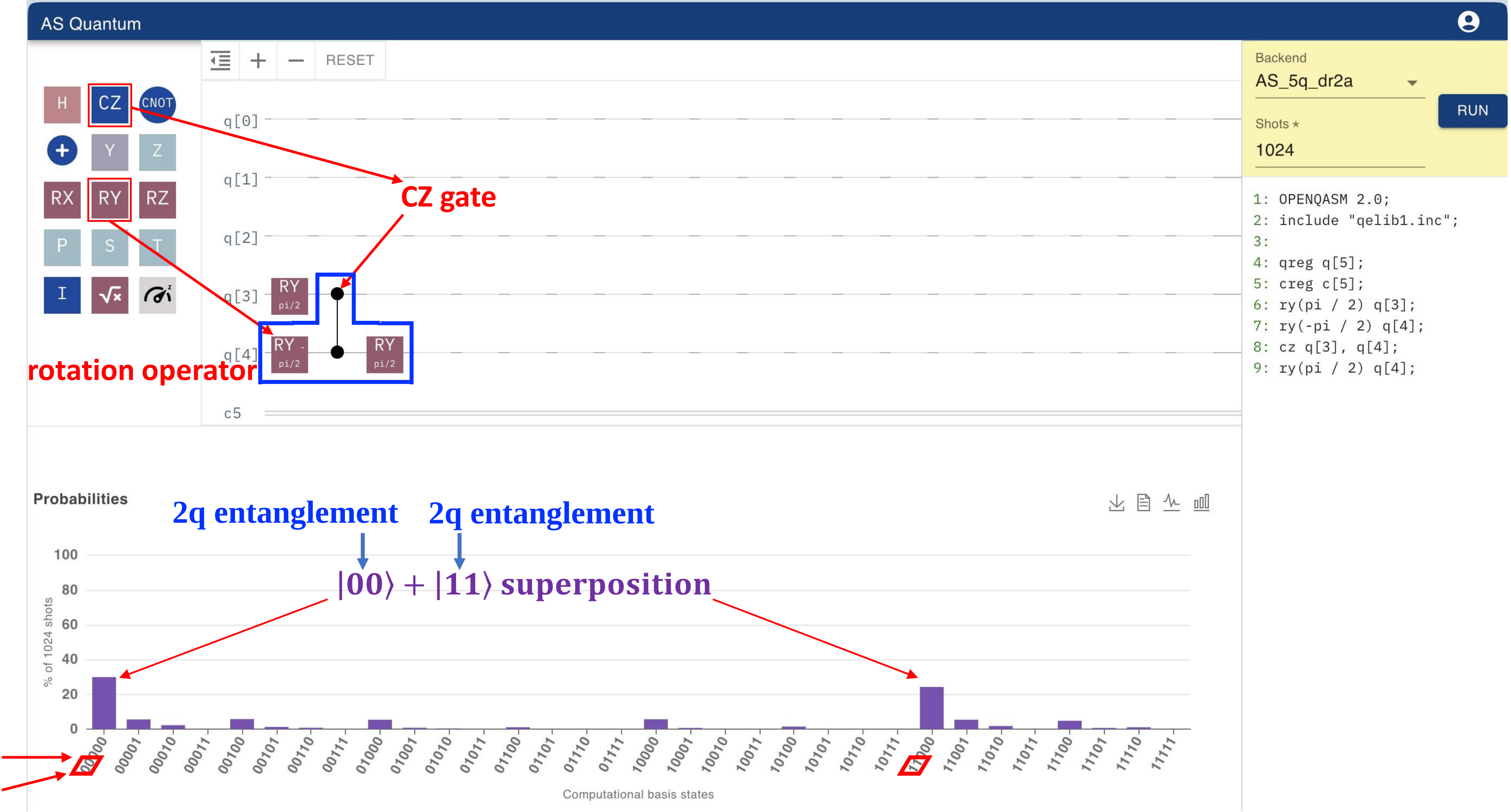
5 qubits

State Probability

$2^5 = 32$ states

q[0]
q[1]
q[2]
q[3]
q[4]

q[3], q[4] entanglement



Backend
AS_5q_dr2a

Shots *
1024

```
OPENQASM 2.0;
include "qelib1.inc";

qreg q[5];
creg c[5];
ry(pi / 2) q[3];
ry(-pi / 2) q[4];
cz q[3], q[4];
ry(pi / 2) q[4];
```

CZ gate

rotation operator

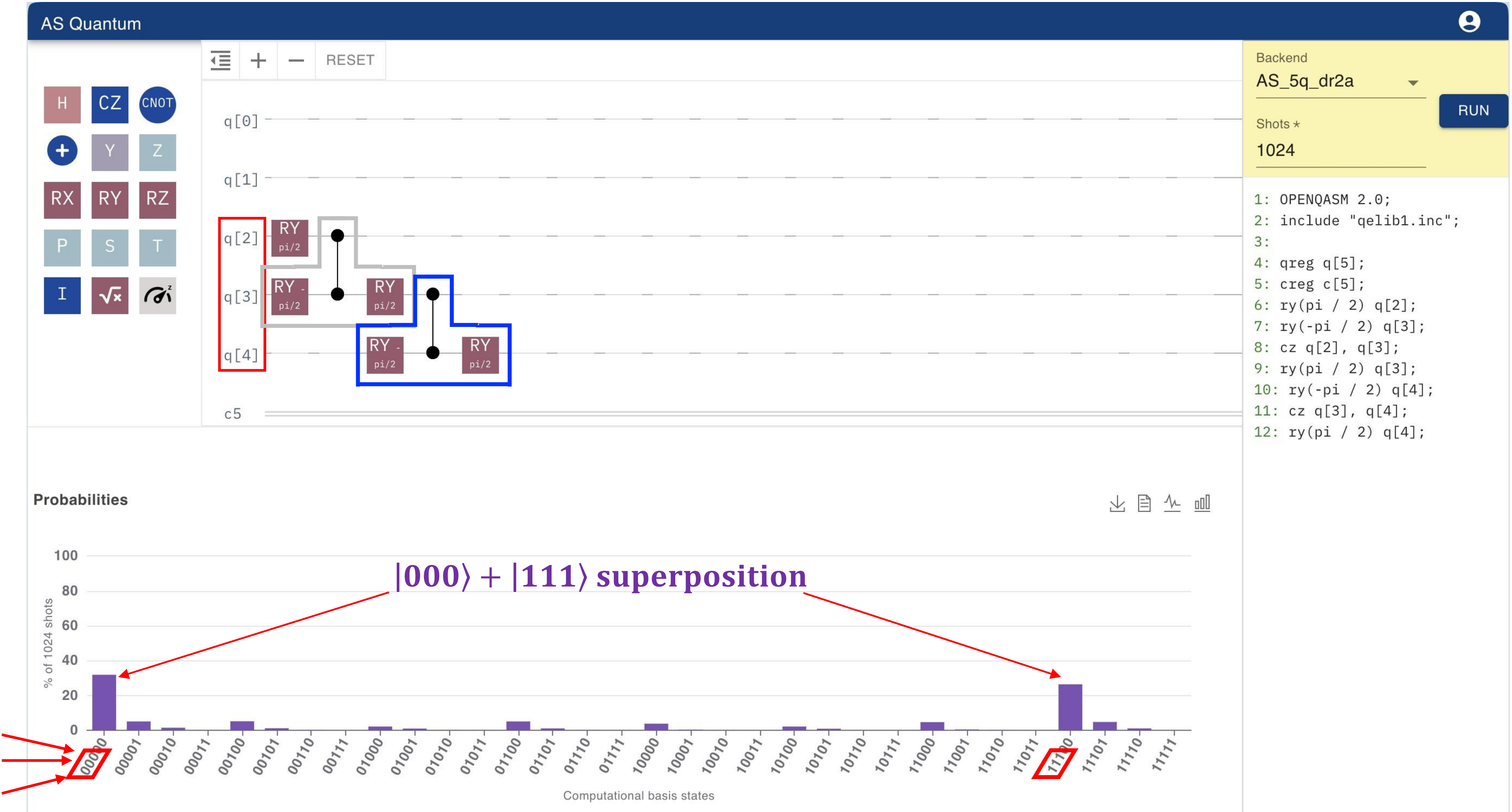
2q entanglement

2q entanglement

$|00\rangle + |11\rangle$ superposition

q[0]
q[1]
q[2]
q[3]
q[4]

q[2], q[3], q[4] entanglement



Backend: AS_5q_dr2a

Shots *: 1024

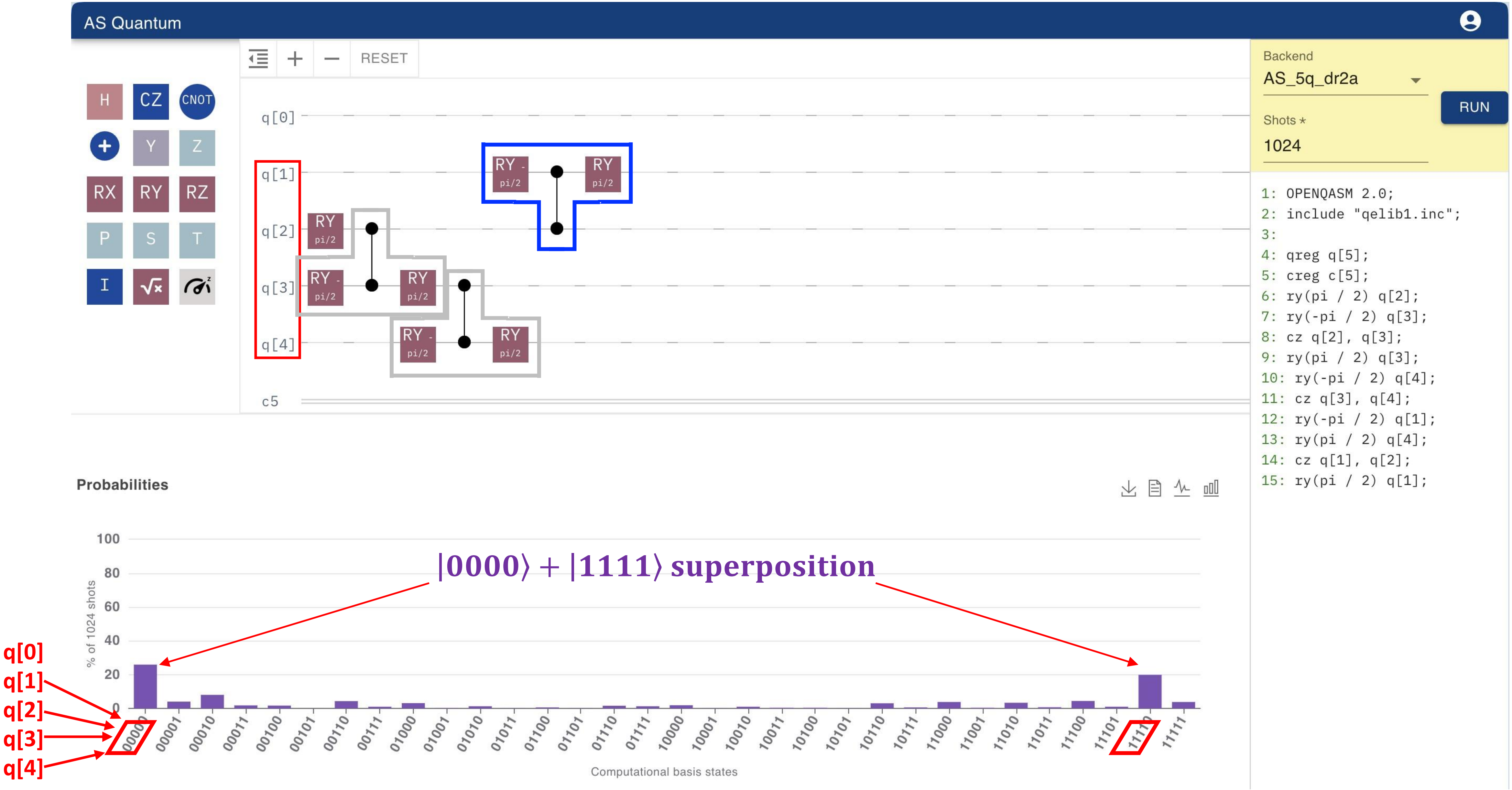
```
OPENQASM 2.0;
include "qelib1.inc";

qreg q[5];
creg c[5];
ry(pi / 2) q[2];
ry(-pi / 2) q[3];
cz q[2], q[3];
ry(pi / 2) q[3];
ry(-pi / 2) q[4];
cz q[3], q[4];
ry(pi / 2) q[4];
```

$|000\rangle + |111\rangle$ superposition

q[0]
q[1]
q[2]
q[3]
q[4]

q[2], q[3], q[4], q[1] entanglement



Backend
AS_5q_dr2a

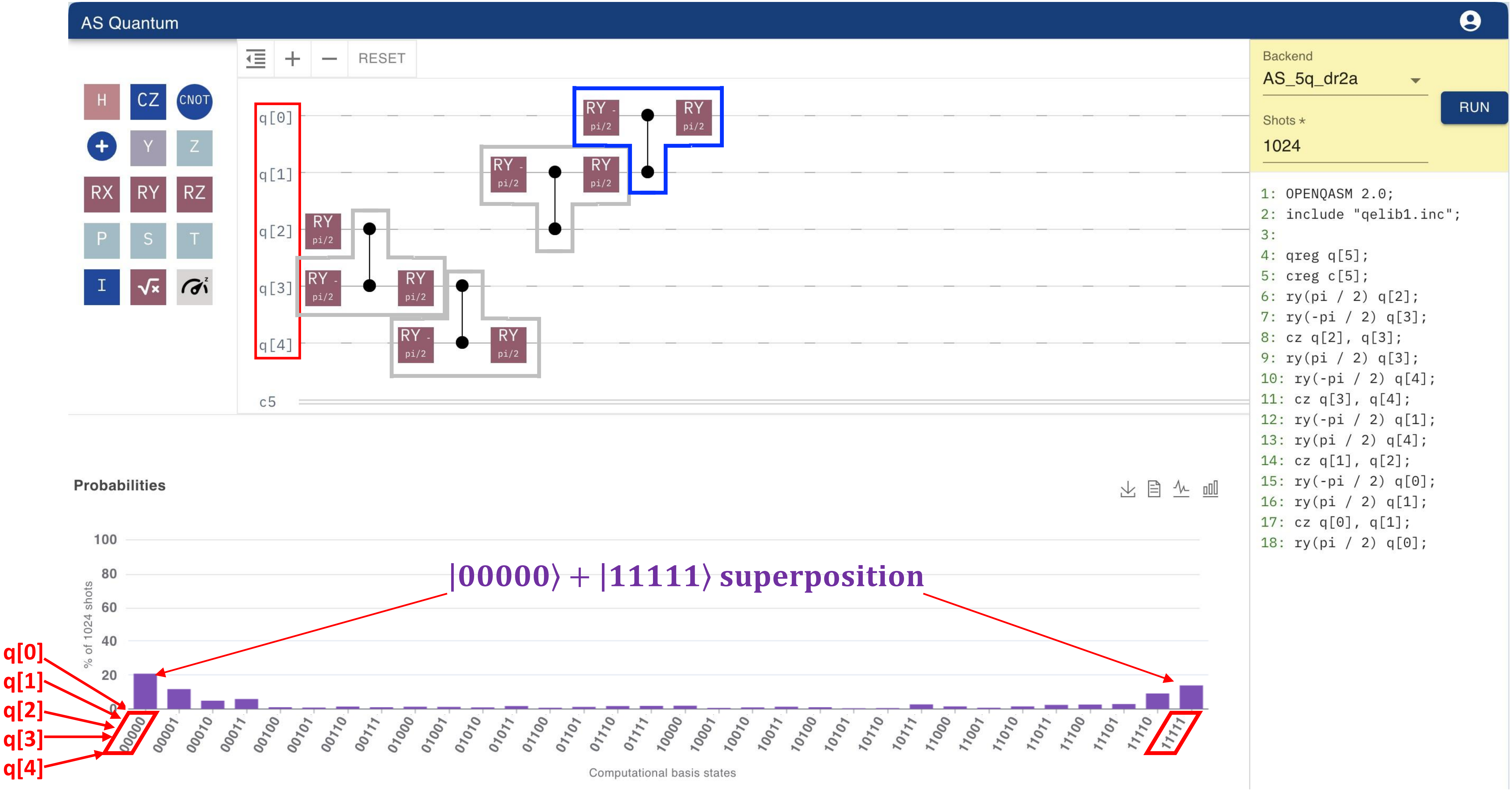
Shots *
1024

```
OPENQASM 2.0;
include "qelib1.inc";

qreg q[5];
creg c[5];
ry(pi / 2) q[2];
ry(-pi / 2) q[3];
cz q[2], q[3];
ry(pi / 2) q[3];
ry(-pi / 2) q[4];
cz q[3], q[4];
ry(-pi / 2) q[1];
ry(pi / 2) q[4];
cz q[1], q[2];
ry(pi / 2) q[1];
```

|0000⟩ + |1111⟩ superposition

q[2], q[3], q[4], q[1], q[0] entanglement



Backend
AS_5q_dr2a

Shots *
1024

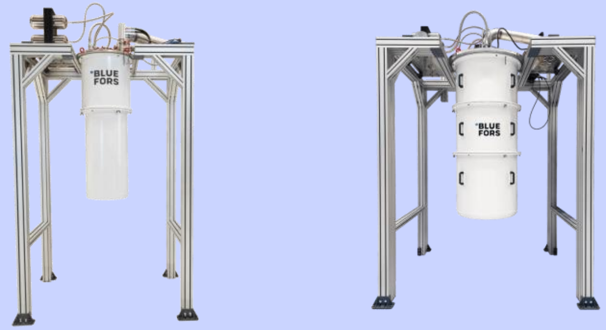
```
OPENQASM 2.0;
include "qelib1.inc";

qreg q[5];
creg c[5];
ry(pi / 2) q[2];
ry(-pi / 2) q[3];
cz q[2], q[3];
ry(pi / 2) q[3];
ry(-pi / 2) q[4];
cz q[3], q[4];
ry(-pi / 2) q[1];
ry(pi / 2) q[4];
cz q[1], q[2];
ry(-pi / 2) q[0];
ry(pi / 2) q[1];
cz q[0], q[1];
ry(pi / 2) q[0];
```

$|00000\rangle + |11111\rangle$ superposition

Building a Quantum Computer

Control Stack



small fridges | **big fridge**

QPU Iteration | QPU Utilization

QPU Betterment

Interleaved RB

QuantrolOX

QuAM

Pulse Scheduler

FPGA

QUA

Control Middleware

QPU Packaging + QPU chips

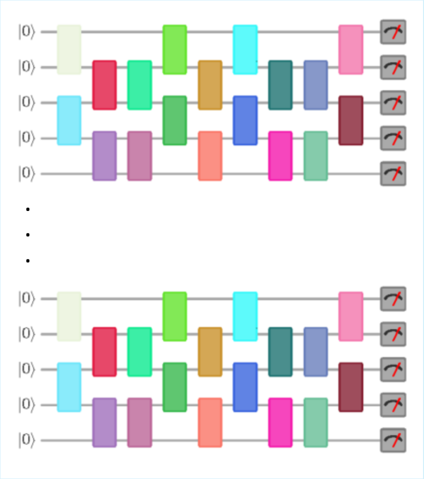
Quantum Operating System

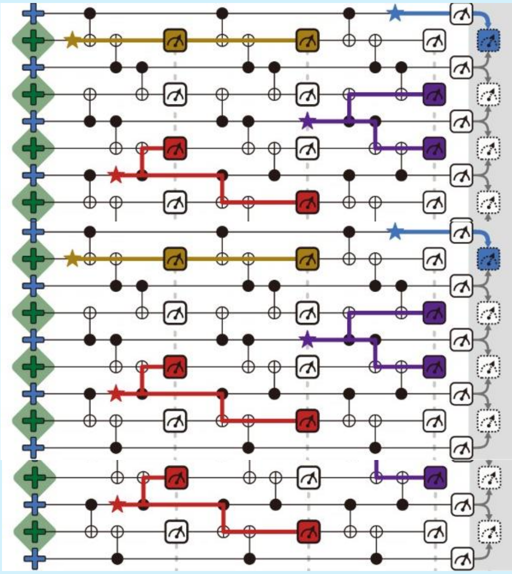
QNN, VQE

Collaborators

Random Quantum Circuits

Repetitive Error Detection



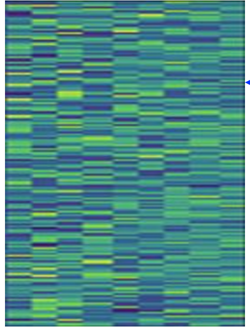


Qiskit & Cirq

Quantum Circuit Compiler & Transpiler

QUM: Backend process manager

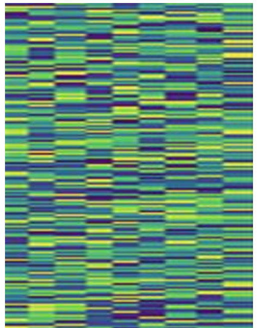
Automated QPU Bring-up, Calibration, Optimization



Cross Entropy Benchmarking

Quantum Utility

Quantum Advantage



CUDA Quantum

Quantum-Classical Programming Platform

Hybrid Applications

Drug Discovery, Chemistry, Weather, Finance, Logistics, and More

High Performance Computing (HPC)

DGX Quantum

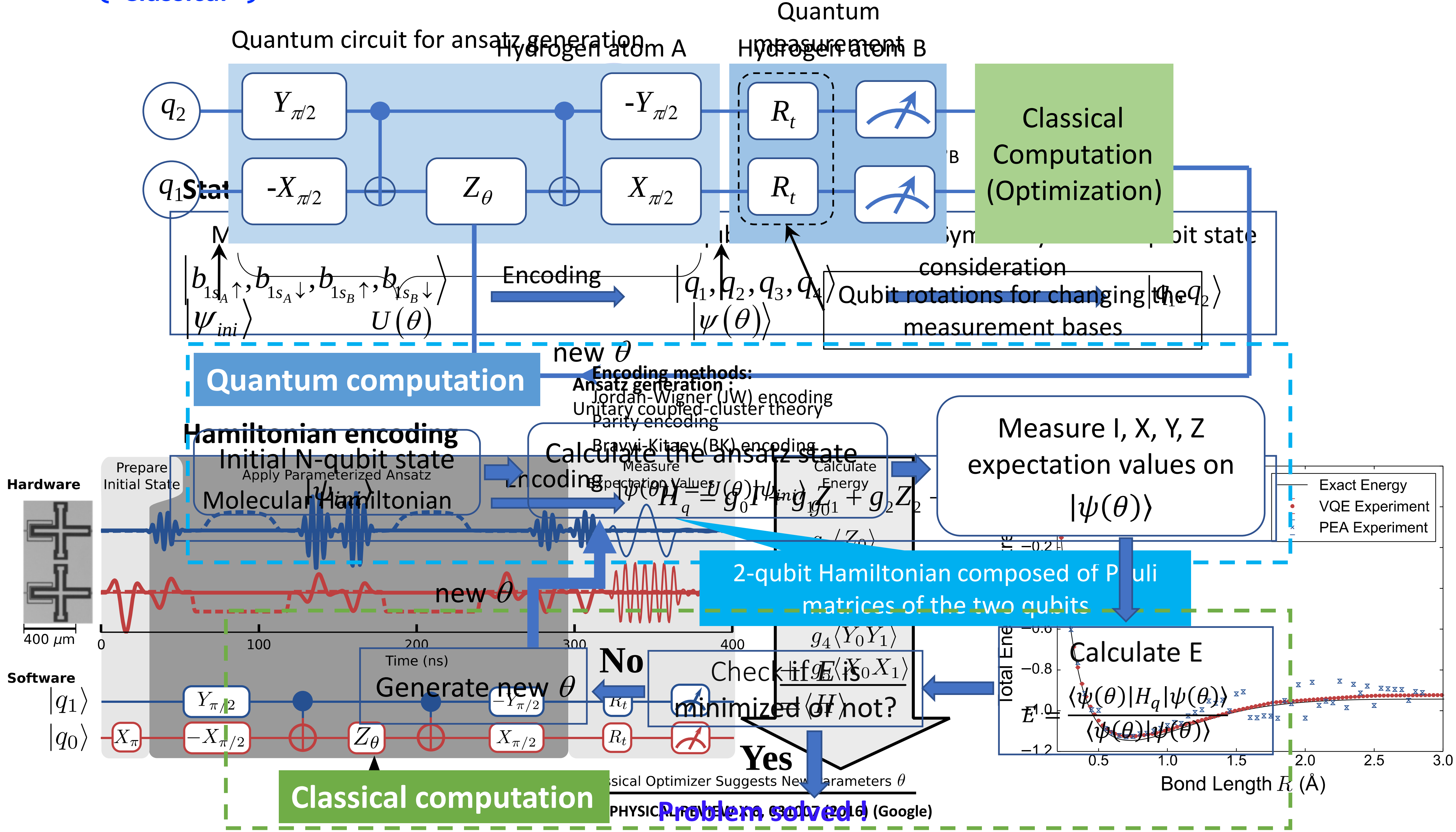
An example of

Classical-Assisted Quantum Computing:

Variational Quantum Eigensolver (VQE)

Hybrid { Quantum / Classical } computation

Variational quantum eigensolver for E_{ground} of the H^2 molecules



PHYSICAL REVIEW X 6, 031007 (2016) (Google)

HPC will always be QC incubator:

Calibrations: improve fidelities and uptime with advanced reinforcement learning optimizers

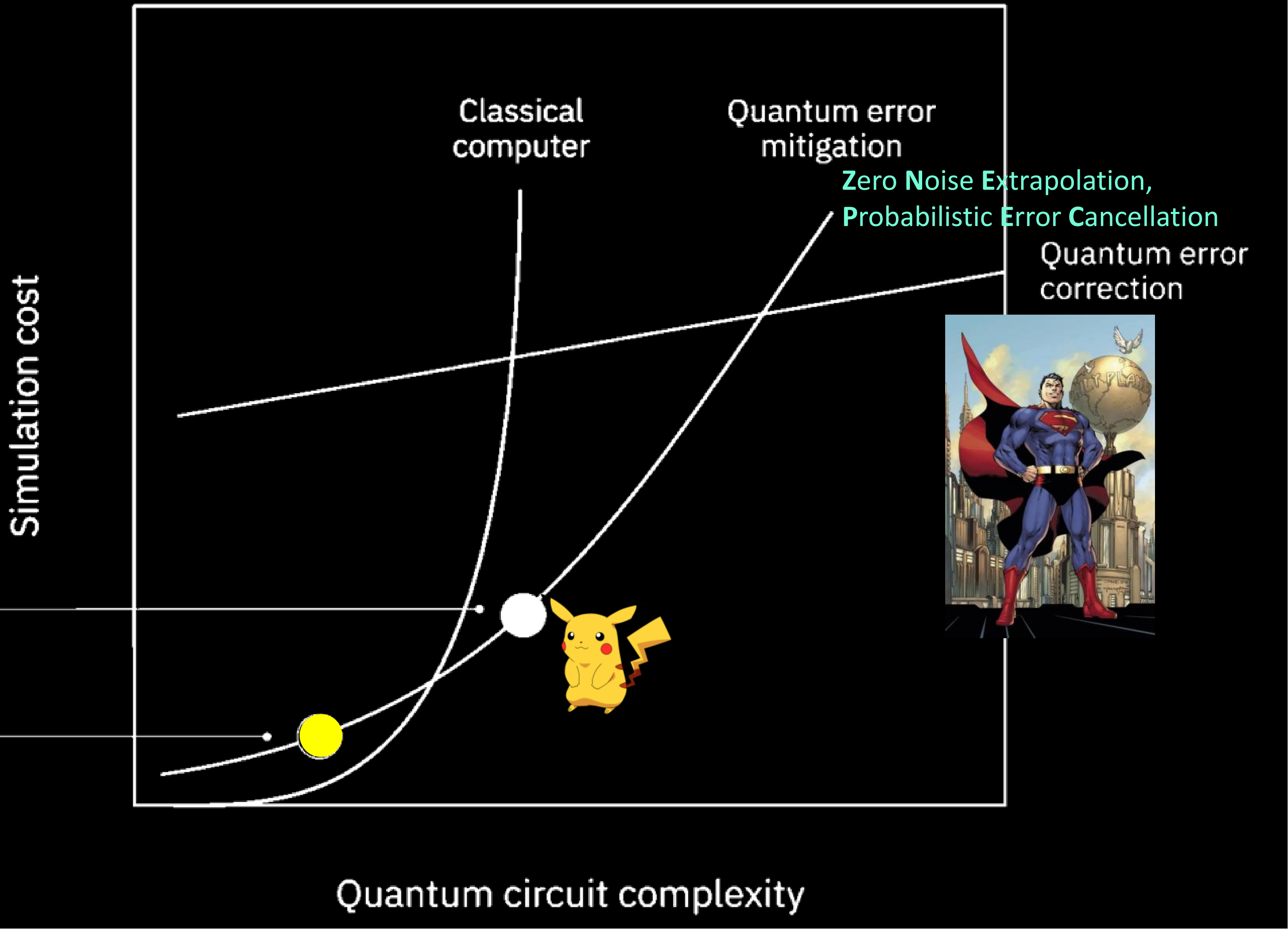
Error mitigation: supercharge classical optimization of control signals

Error correction: implement powerful decoders in the GPU accessible in <4 usec

Hybrid applications: Make it easier to leverage the QPU as an accelerator for subroutines in larger workflows – cut down QPU-CPU latency

We want to get here

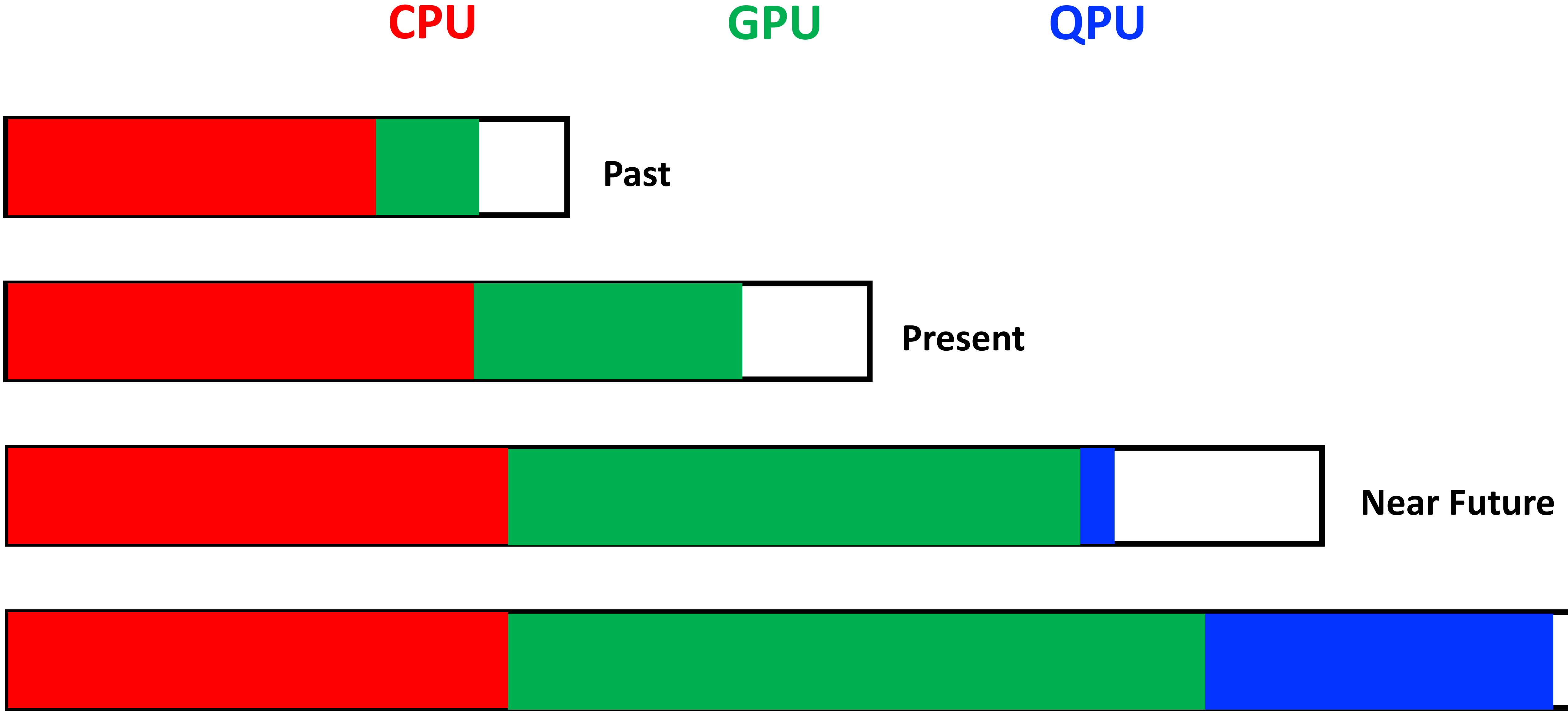
Today we are here



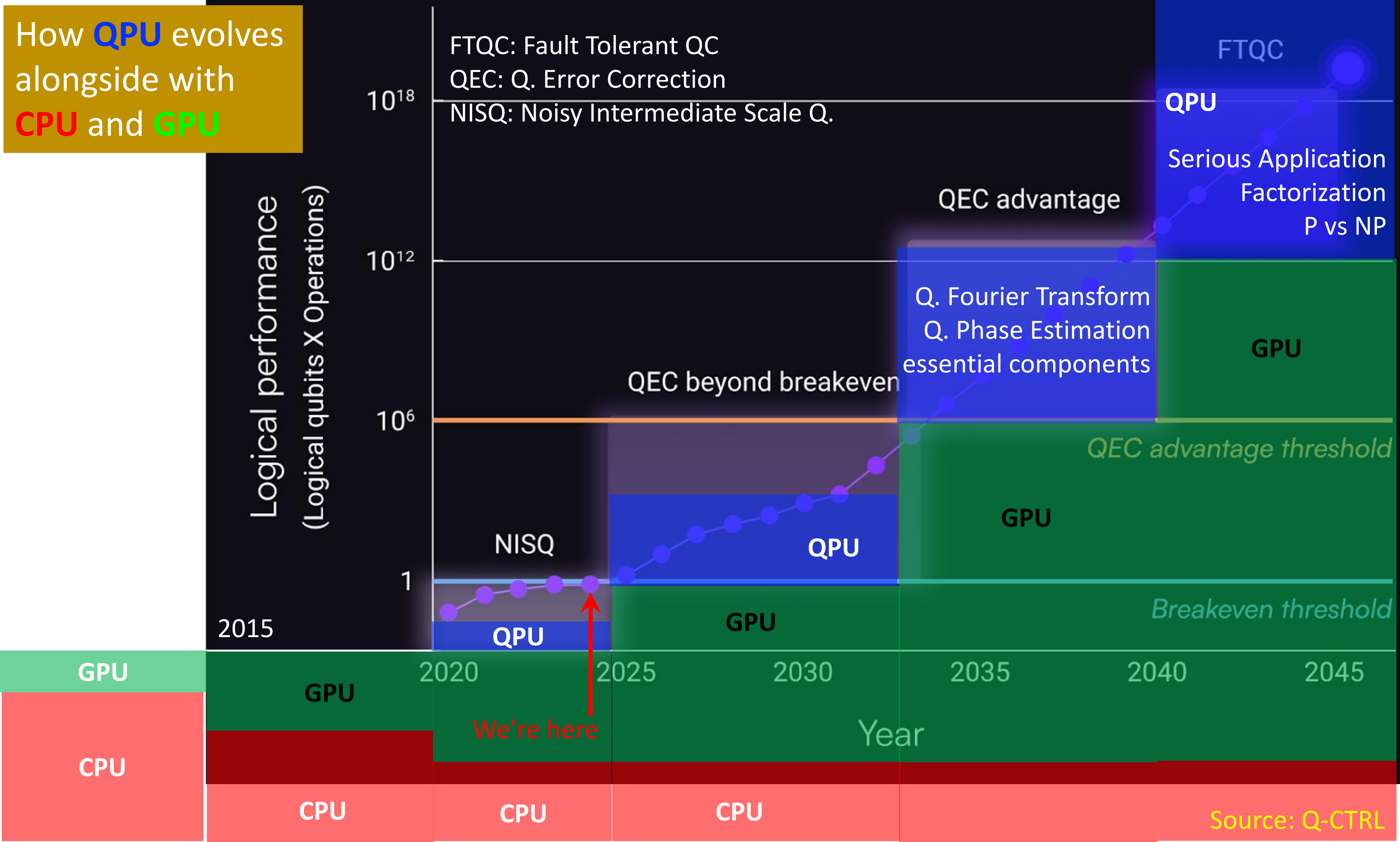
Source: IBM, QM

Take home message

Data Center



How **QPU** evolves alongside with **CPU** and **GPU**



FTQC: Fault Tolerant QC
QEC: Q. Error Correction
NISQ: Noisy Intermediate Scale Q.

Source: Q-CTRL

Google Launches $5 Million XPRIZE to Find Real World Uses for Quantum Computers

March 5, 2024

We expect that competitive submissions will make at least one of the following types of contributions (we also give a few examples from the last five years; however, note that these examples are not in any way an expression of preferred areas of focus):

1. A new quantum algorithm for solving a new class of problems with quantum advantage.

Example: quartic quantum speedup for tensor principal component analysis (arXiv: 1907.12724). Submission would be incomplete without suggesting a target real-world application and submission would be much stronger with some estimatedresources for quantum advantage. Still, significant points for novelty.

2. Work showing how existing quantum algorithms can be used to solve previously unknown applications with a quantum advantage.5

Example 1: using quantum linear system solvers or Hamiltonian simulation to give super-quadratic speedup in simulating classical waves (arXiv:1711.05394) or coupled harmonic systems (arXiv:2303.13012). Submissions would be stronger with some estimated resources for quantum advantage in real-world applications.

Example 2: using quantum simulation to better design fusion reactors (arXiv:2308.12352). Weakness is that quantum simulation applications are not especially hard to find and resources required for advantage are still fairly high.

3. Work significantly reducing the resources required for a quantum computer to reach quantum advantage for an already established algorithm/application.

Example 1: improved chemistry algorithms (arXiv:2011.03494) with application to simulating the FeMoCo nitrogen fixation catalyst. Submission would be stronger if the magnitude of the resource reduction and thought delta were larger.

Example 2: improved algorithms for topological data analysis (arXiv:2209.13581 and arXiv:2209.12887). A significant weakness is that neither paper identifies real-world occurrences of the problem where quantum advantage is viable.

https://assets-us-01.kc-usercontent.com/5cb25086-82d2-4c89-94f0-8450813a0fd3/be438f12-70ca-42e6-a381-30ffb52031c2/XPRIZE%20Quantum%20Applications%20Preliminary%20Guidelines_V.01.pdf https://developer.nvidia.com/blog/an-introduction-to-quantum-accelerated-supercomputing/

It is a common misconception that QPUs will accelerate any sort of computation. This will probably not be the case, as QPUs are well-suited for very specific tasks. One of the primary weaknesses of a quantum computer is the fact that information can only be extracted via nondeterministic measurements of the N qubits to produce a bitstring of length N. Therefore, it is important to understand the types of problems that are either theoretically proven or expected to have efficient implementations on a QPU, as listed below.

- **Simulating quantum systems:** QPUs, quantum systems themselves, are naturally good at simulating other quantum systems. This could enable all sorts of fundamental science ranging from the exploration of new chemical reactions and materials to unlocking the mysteries of high-energy physics.
- **Optimization:** The exponential amount of information held in a QPU could allow for new methods aimed at finding better solutions to large combinatorial optimization problems, benefitting diverse use cases including route planning, grid optimization, genetics, and portfolio selection.
- **AI and machine learning:** The properties of QPUs make them amenable for building and sampling from complex distributions and deploying novel methods for finding patterns in high-dimensional data sets. These techniques could be highly portable and benefit almost any domain of science and industry.
- **Monte Carlo estimation:** QPUs can obtain a theoretical quadratic speedup for Monte Carlo estimation, which would improve the accuracy and speed of obtaining risk metrics and financial predictions critical for getting an edge in the markets.
- **Fluid dynamics:** Aerodynamic, weather, and reservoir simulations are examples of multiscale problems where systems of differential equations need to be solved with extreme precision using a large grid. QPUs are being explored as tools for solving systems of differential equations that enable far more precise fluid dynamic simulations.

The high-performance platform for hybrid quantum-classical computing

NVIDIA CUDA-Q

Key Benefits

Productive

Streamlines hybrid quantum-classical development with a unified programming model, improving productivity and scalability in quantum algorithm research.

Flexible Platform

Connects to partner QPUs and GPU simulators, easy toolchain integration, and interoperates with modern GPU-accelerated applications.

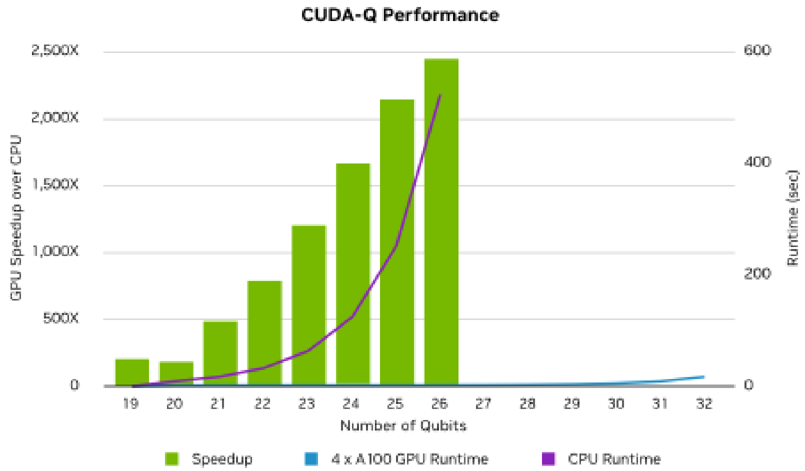
High Performing

2500X simulation speedup on a four A100 GPU for up to 26 qubits, and scaling to 40 qubits by distributing the simulation across 128 GPU nodes.

Built for Performance

NVIDIA CUDA-Q enables straightforward execution of hybrid code on many different types of quantum processors, simulated or physical. Researchers can leverage the cuQuantum-accelerated simulation backends as well as QPUs from our partners or connect their own simulator or quantum processor.

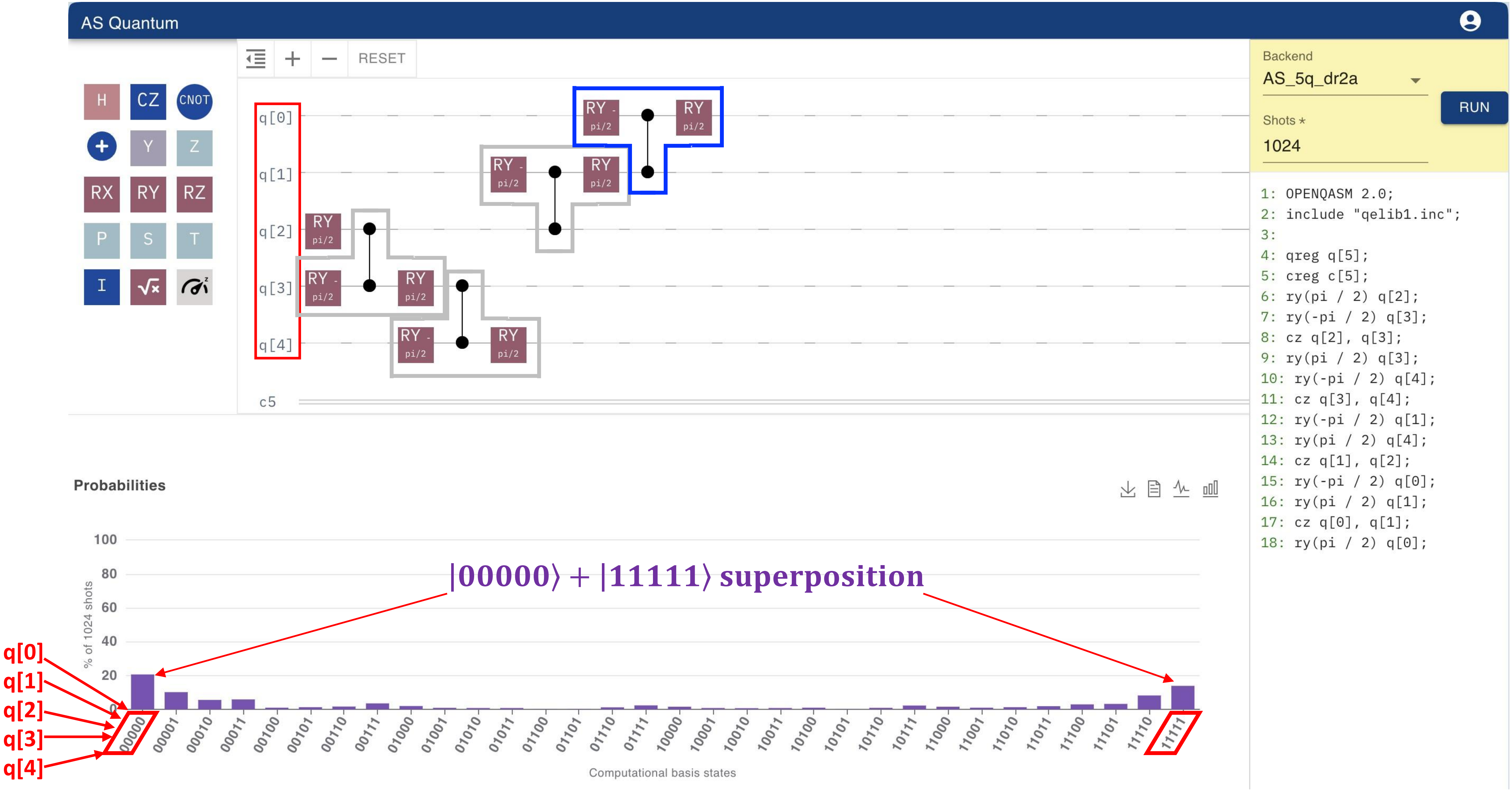
NVIDIA CUDA-Q can significantly speed up quantum algorithms, compared to other quantum frameworks. Quantum algorithms can achieve a speedup of up to 2500X over CPU, scaling number of qubits using multiple GPUs.



Typical QML workflow in CUDA-Q using multi-threaded CPU versus multiple NVIDIA A100 Tensor Core GPUs.

https://developer.nvidia.com/cuda-q

q[2], q[3], q[4], q[1], q[0] entanglement



Backend: AS_5q_dr2a

Shots *: 1024

```
OPENQASM 2.0;
include "qelib1.inc";

qreg q[5];
creg c[5];
ry(pi / 2) q[2];
ry(-pi / 2) q[3];
cz q[2], q[3];
ry(pi / 2) q[3];
ry(-pi / 2) q[4];
cz q[3], q[4];
ry(-pi / 2) q[1];
ry(pi / 2) q[4];
cz q[1], q[2];
ry(-pi / 2) q[0];
ry(pi / 2) q[1];
cz q[0], q[1];
ry(pi / 2) q[0];
```

$|00000\rangle + |11111\rangle$ superposition