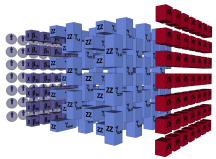
Google AI Quantum

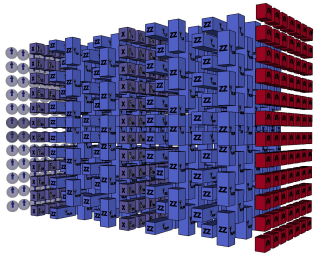
Cirq

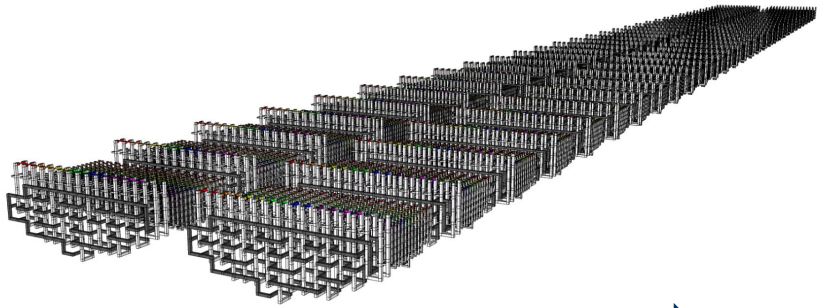
Dave Bacon (dabacon@google.com)

May 13, 2019

A Privileged Age







Supremacy Frontier

we are here

100-1000 qubits
3-4 9's of fidelity
(NISQ)

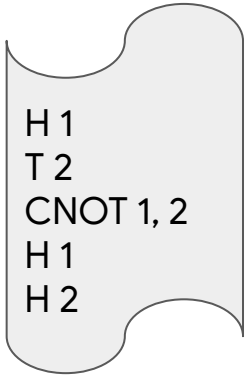
~10^6 qubits
well below threshold
= ~10^3 error corrected qubits

Are there impactful algorithms for noisy intermediate scale quantum (NISQ) processors?

Google AI Quantum

Quantum Software

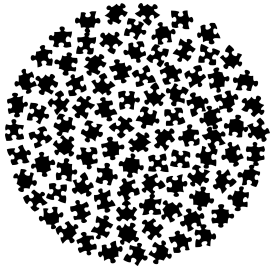
Fidelity: 90 99 99.9 99.99 ..... ∞9s



Write it on a piece of paper!

Useful to record instructions.

Can still eyeball circuits.



At supremacy frontier.

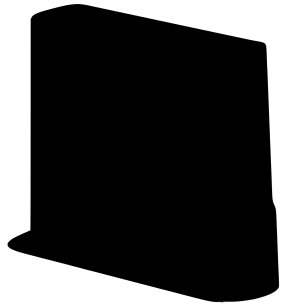
Depth and gate minimization.

Simple modularity.

Complex modularity.

Automatic compiling.

Beginning of hardware independent abstractions.



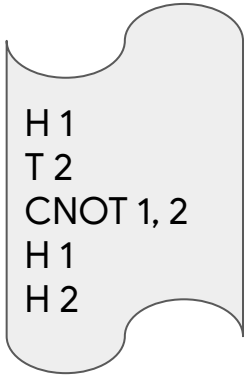
Architecture.

Operating systems.

High level languages.

Quantum Software

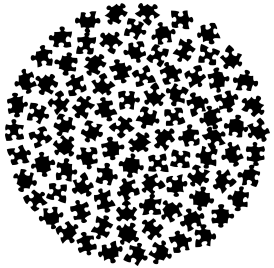
Fidelity: 90 | 99 | 99.9 | 99.99 | ..... | ∞9s



Write it on a piece of paper!

Useful to record instructions.

Can still eyeball circuits.



At supremacy frontier.

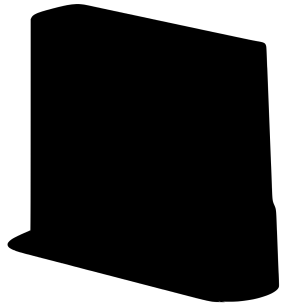
Depth and gate minimization.

Simple modularity.

Complex modularity.

Automatic compiling.

Beginning of hardware independent abstractions.



Architecture.

Operating systems.

High level languages.

An open source Python framework for
Noisy Intermediate Scale Quantum (NISQ) algorithms

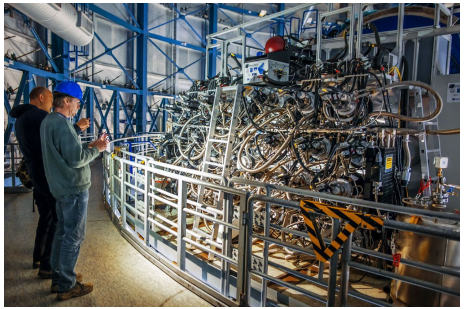
https://github.com/quantumlib/Cirq

Out of the dark ages

Wizards

Out of the dark ages

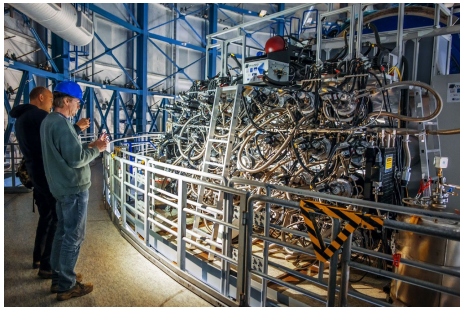
Wizards



Experiments

Out of the dark ages

Wizards



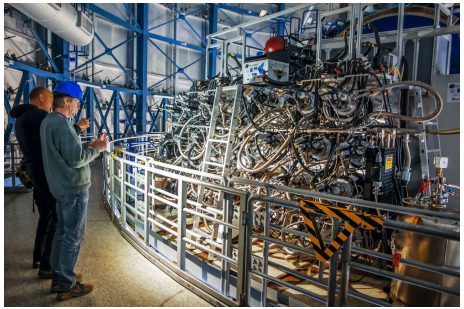
Experiments

Muggles

Google AI Quantum

Out of the dark ages

Wizards



Experiments

Muggles

```
import cirq

# Define a qubit.
qubit = cirq.NamedQubit("a")

# Build a simple quantum circuit.
circuit = cirq.Circuit.from_ops(
    cirq.X(qubit)**(0.5),  # Square root of NOT
    cirq.measure(qubit)  # Measurement.
)

print(circuit)
```

```
a: ———X^0.5———M———
```

Programs

Google AI Quantum

Cirq Core Philosophies

- Hardware details need to be part of programming abstractions as they greatly impact the viability of NISQ algorithms.
- Hardware should drive features and diverse hardware will have diverse features.
- Data structures and abstractions should match context in which they are used (optimization, simulation, execution).
- Optimize for workflows that validate heuristics algorithms and for rapid iteration in exploring minimally sized circuits.

Deutsch-Jozsa

x	f(x)
0	0
1	0

x	f(x)
0	1
1	1

Constant

x	f(x)
0	0
1	1

x	f(x)
0	1
1	0

Balanced

Deutsch-Jozsa

x	f(x)
0	0
1	0

x	f(x)
0	1
1	1

x	f(x)
0	0
1	1

x	f(x)
0	1
1	0

Constant

Balanced

x	y	x	f(x)⊕y
0	0	0	0
0	1	0	1
1	0	1	0
1	1	1	1

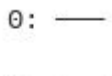
x	y	x	f(x)⊕y
0	0	0	1
0	1	0	0
1	0	1	1
1	1	1	0

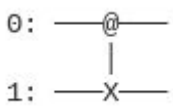
x	y	x	f(x)⊕y
0	0	0	0
0	1	0	1
1	0	1	1
1	1	1	0

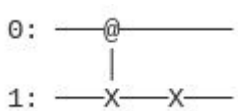
x	y	x	f(x)⊕y
0	0	0	1
0	1	0	0
1	0	1	0
1	1	1	1

Google AI Quantum

Deutsch-Jozsa







Constant

x	y	x	f(x)⊕y
0	0	0	0
0	1	0	1
1	0	1	0
1	1	1	1

x	y	x	f(x)⊕y
0	0	0	1
0	1	0	0
1	0	1	1
1	1	1	0

Balanced

x	y	x	f(x)⊕y
0	0	0	0
0	1	0	1
1	0	1	1
1	1	1	0

x	y	x	f(x)⊕y
0	0	0	1
0	1	0	0
1	0	1	0
1	1	1	1

Phase Kickback

$$U_f|x\rangle|y\rangle = |x\rangle|y \oplus f(x)\rangle$$

$$U_f|x\rangle\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) = \frac{1}{\sqrt{2}}|x\rangle(|f(x)\rangle - |1 \oplus f(x)\rangle))$$

$$= (-1)^{f(x)}|x\rangle\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$$

Fin

Google AI Quantum