Physics 770 Seminar

Computing, Quantum

Room CP 297, 2-3PM, April 3, 2026

Prof. Henry Dietz
Electrical & Computer Engineering

Aggregate.Org
UNBRIDLED COMPUTING

University of Kentucky

Abstract

In November 2024, El Capitan officially became the world's fastest supercomputer. It claimed that record using 11,039,616 cores and about 30MW. About a month later, Google Quantum AI announced that their Willow quantum computer chip "performed a standard benchmark computation in under five minutes that would take one of today’s fastest supercomputers 10 septillion (that is, 10^25) years." One could upscale a supercomputer like El Capitan in obvious ways to match the performance claimed by Willow, but that machine would require the energy output of billions of Suns! The catch is that most computations that you can do in under five minutes on a $3 microcontroller are not even theoretically possible on a Willow chip...

This talk is not about quantum physics, but about why people are investing so much in trying to use quantum phenomena for computation and how such machines work as computers.

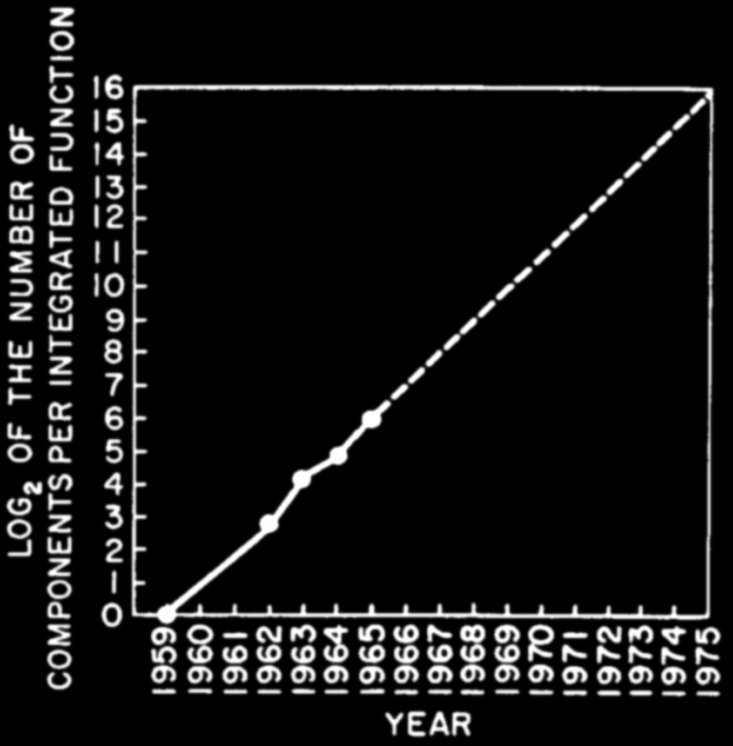
University of Kentucky

Quantum Phenomena in Computers

- Implement conventional logic gates, bit storage, interconnection networks, etc.
- **Integrated circuits use quantum phenomena**
  - MOSFET transistors based on bandgap
  - Doping to control quantum material properties
  - Quantum tunneling to write EEPROM and a key constraint in nm-scale transistor design
  - Electroluminescence in LEDs & laser diodes

How Computers Get Faster: Moore's Law

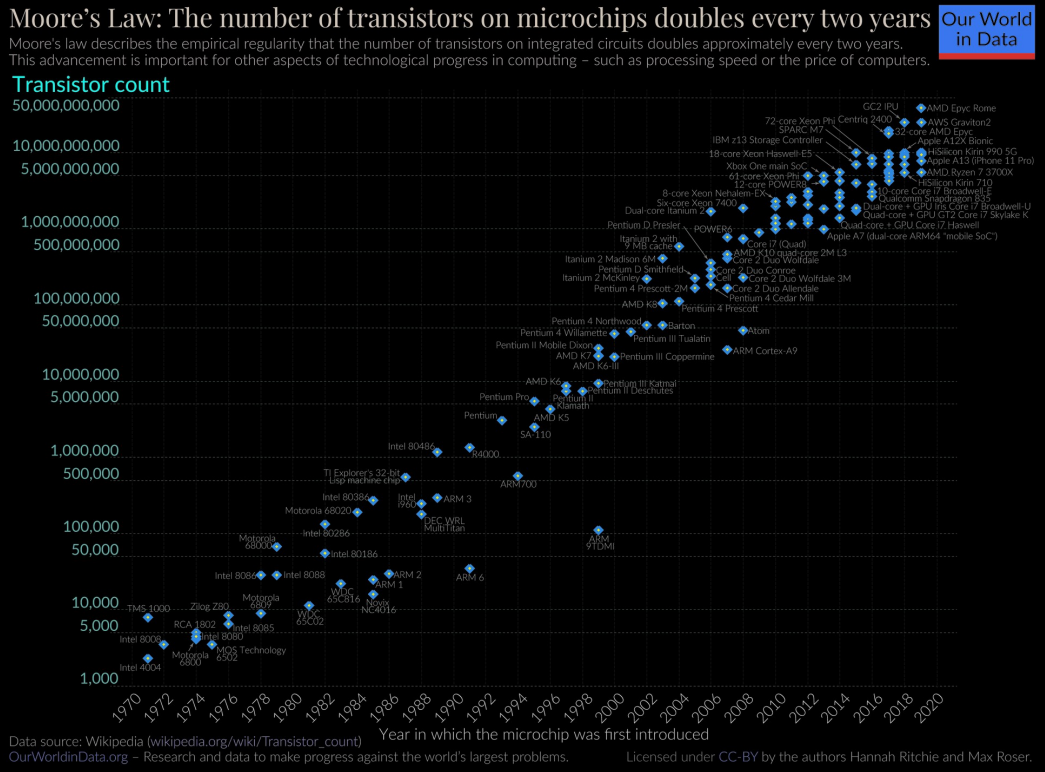
- 1965 prediction
  - Not about chip speed
  - Circuit complexity 2X every 18-24 months
- Speedup is mostly about **parallel processing**



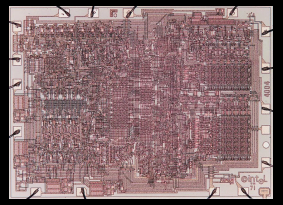
Parallel Processing

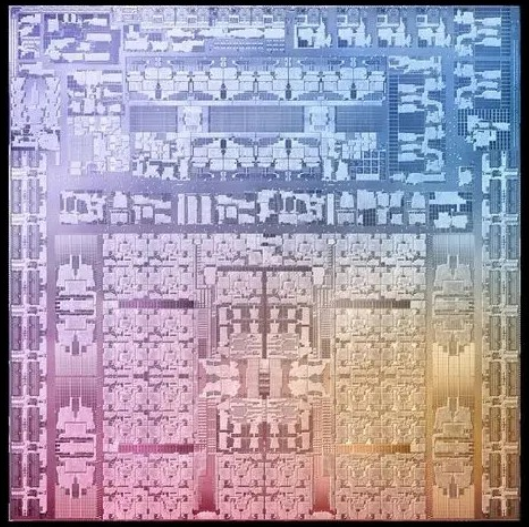
- Break program into ***N*** pieces that can execute simultaneously
  - **Scalable**: bigger ***N***, more speedup
  - **Modular hardware**
  - Can be **fault tolerant** using redundancy
- This scales up forever, right?

Moore’s Law is still sort-of OK...

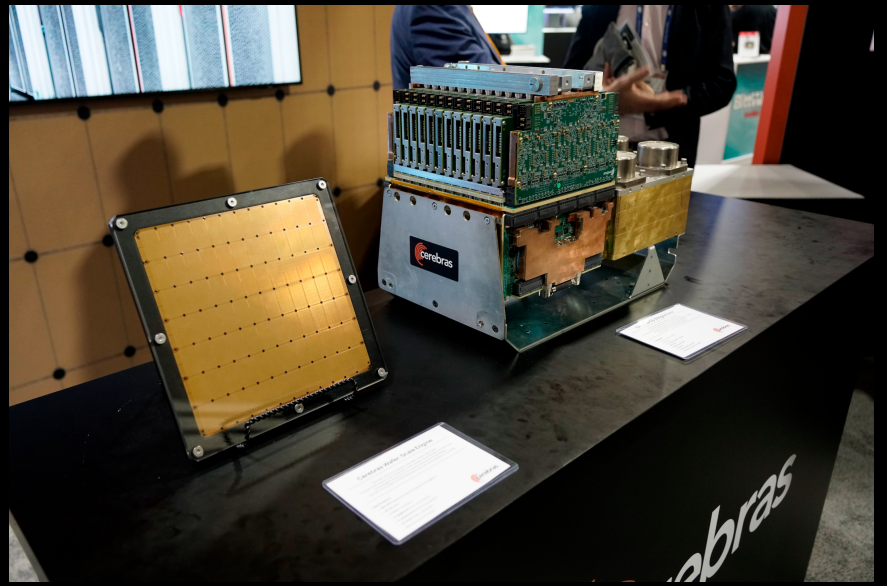


Dies





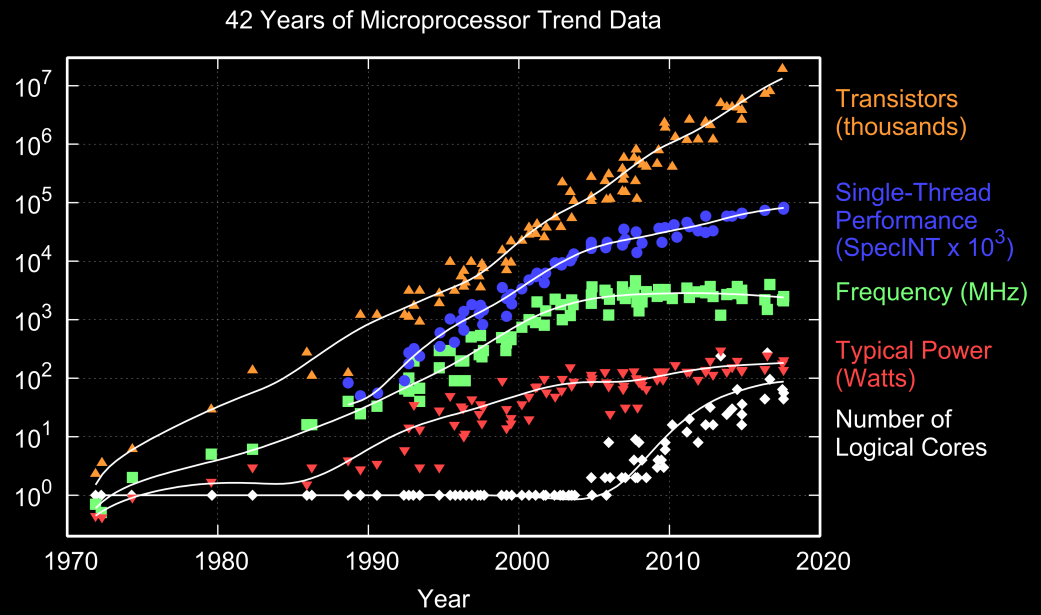
M3 Max (M3 Ultra connects two)



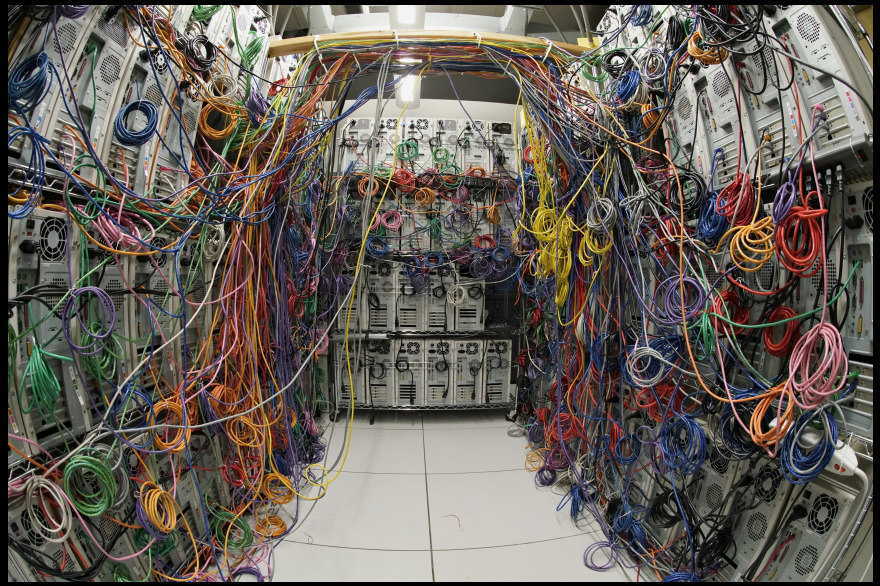
- **1971:** Intel 4004: 2,300 trans, 10µm, 12mm^2
- Apple M3 Ultra: 184G trans, 3nm, 2x600mm^2
- Cerebras WSE-3: 4T trans, 5nm, 46225mm^2

All The Bad News

- Moore's Law slowing
- Power/transistor ▼ slower than transistors/chip ▲
- Individual ops not getting much faster



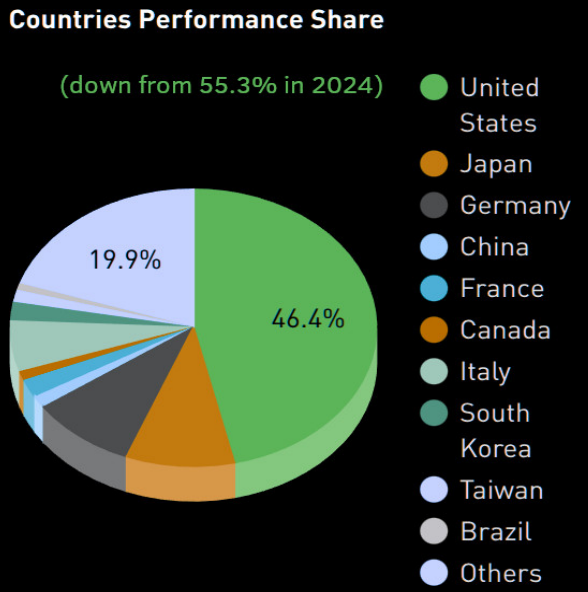
108A Marksbury (my lab's machine room)



- **175 KW** and **30 Tons** cooling
  (waste heat heats half the building)

Top500 @ SC25

St. Louis, MO | hpc ignites.



- $600M **El Capitan** @ Lawrence Livermore National Laboratory is ***still*** #1
- HPE Cray EX255a using 11,039,616 cores using **AMD** 24-core EPYC + MI300A
- HPL performance is 1.809 Exaflop/s with 2.79 theoretical peak, using **~35MW**
- **Frontier**'s 8,699,904 **AMD** cores with HPL 1.353 Exaflop/s is ***still*** #2

Plumbing @ SC25

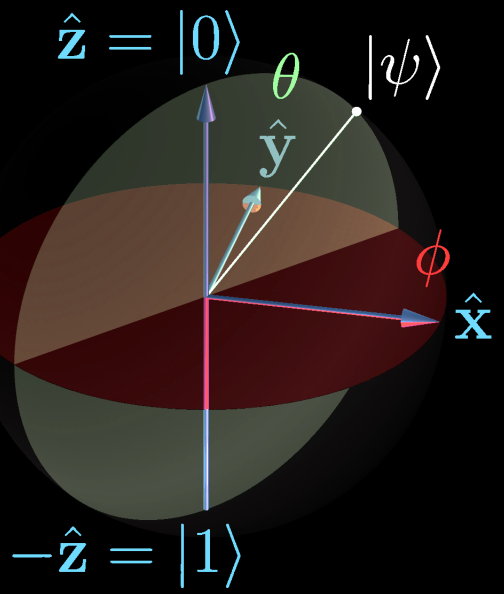
St. Louis, MO | hpc ignites.

- **AI datacenter systems are MUCH larger!**
- Many being built that each will draw **1-5 GW**

Why Look To Quantum Computing?

- Massively-parallel processing ***without*** massively parallel hardware
- Potentially very low power consumption per unit computation performed
- Speedup not driven by Moore's Law; some quantum *algorithms* are exponentially faster
- Different programming models, new algorithms

Bloch Sphere Qubit Model



$$|\psi\rangle = \cos(\theta/2)|0\rangle + e^{i\phi}\sin(\theta/2)|1\rangle$$

$$= \cos(\theta/2)|0\rangle +$$

$$(\cos\phi + i\sin\phi)\sin(\theta/2)|1\rangle$$

where $0 \leq \theta \leq \pi$ and $0 \leq \phi < 2\pi$

- Visualizes **wave function** of ***a single*** qubit
- Probability 0/1 by coordinates on sphere surface

But Where's The Parallelism?

- Bloch sphere is one qubit in **superposition**
- Multiple qubits can be **entangled** so that ***E* Qubits** hold a ***probability density function*** over **all 2^E-bit values**
  - Each qubit holds up to 2^E bits
  - One operation on one qubit gives 2^E results

Quantum Computing

- State-of-the-art conventional processor chips already depend on quantum phenomena...
- **Quantum Computing** is about ***implementing a model*** that leverages quantum concepts
  - **Quantum anealing** (Adiabatic optimization)
  - **Quantum inspired** (**QI**)
  - **Quantum circuits / gates**

Quantum Annealing

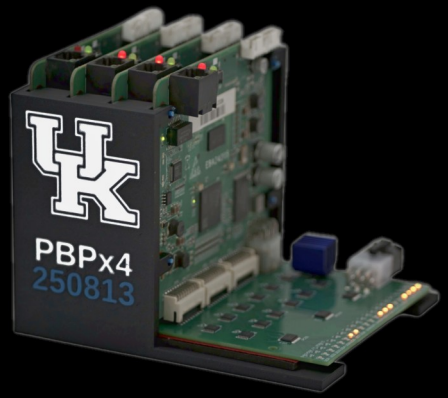
QUBIT	q_i	**Quantum bit** which participates in annealing cycle and settles into one of two possible final states: $\{0,1\}$
COUPLER	$q_i q_j$	Physical device that allows one **qubit to influence another qubit**
WEIGHT	a_i	Real-valued **constant associated with each qubit**, which influences the qubit's tendency to collapse into its two possible final states; controlled by the programmer
STRENGTH	b_{ij}	Real-valued **constant associated with each coupler**, which controls the influence exerted by one qubit on another; controlled by the programmer
OBJECTIVE	Obj	Real-valued **function which is minimized** during the annealing cycle

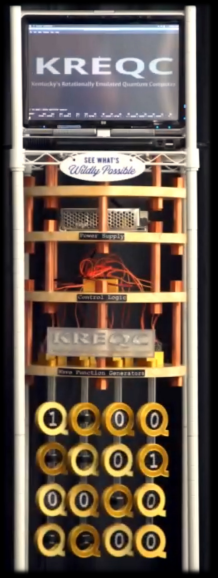
The system samples from the q_i that minimize the objective

$$Obj(a_i, b_{ij}; q_i) = \sum_i a_i q_i + \sum_{ij} b_{ij} q_i q_j$$

Quantum Inspired (QI)

- Does not directly use quantum phenomena
  - Different implementation
  - Uses quantum concepts
- **Parallel Bit Pattern** (**PBP**) represents superpostions as compressed bit patterns





Quantum Circuit Model

- Quantum circuits are equivalent to combinatorial logic – not a state machine due to decoherence, hence not “Turing complete”
- Each reversible operation is equivalent to multiplying the value by a **unitary matrix**, which is essentially a set of rotations; can “uncompute”
- Measurement collapses a superposition

Matrix Representations

- A single bit value is $|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ or $|1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$
- A single qubit is: $|a\rangle = v_0|0\rangle + v_1|1\rangle \rightarrow \begin{bmatrix} v_0 \\ v_1 \end{bmatrix}$
- The value of a pair of qubits is:

$$|\psi\rangle = v_{00}|00\rangle + v_{01}|01\rangle + v_{10}|10\rangle + v_{11}|11\rangle \rightarrow \begin{bmatrix} v_{00} \\ v_{01} \\ v_{10} \\ v_{11} \end{bmatrix}$$

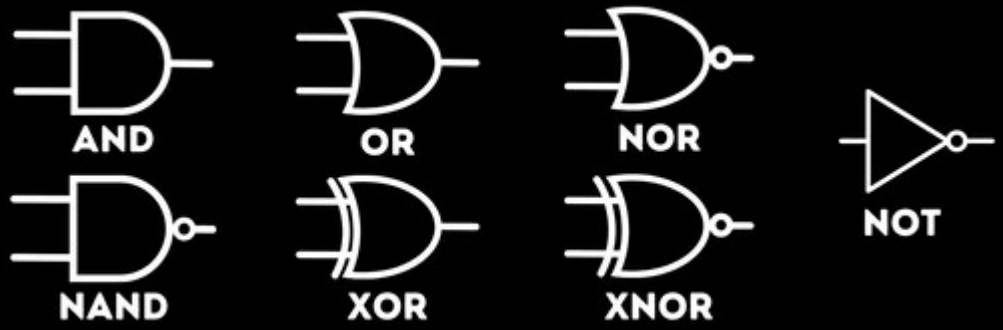
Unitary Matrix Operations

- A unitary matrix is a complex square matrix whose inverse is its conjugate transpose
- Each gate acting on *n* qubits is 2^n x 2^n
- Gate A followed by B is the same as gate B·A
- **Tensor**/**Kronecker** product for gates in parallel
- Any unitary can be converted to a set of gates each acting on either one or two qubits
- Measurement is not unitary, not reversible

Conventional Computing

- **Memory** is made of **Bits**, each holding 0 or 1
  - Bit values **reliably** persist ***forever***
  - **Every bit** can be accessed by addressing
- **Processor** (perhaps one of many in a system)
  - Gates: **`AND`**, **`OR`**, **`XOR`**, **`NOT`**, **`NAND`**, **`NOR`**, **`MUX`**…
  - **Fanout** is allowed (e.g., FOF = fanout of 4)

Conventional Logic Gates



NOT gate

A	$\overline{A}$
0	1
1	0

Inputs		Outputs					
A	B	AND	NAND	OR	NOR	XOR	XNOR
0	0	0	1	0	1	0	1
0	1	0	1	1	0	1	0
1	0	0	1	1	0	1	0
1	1	1	0	1	0	0	1

- Inputs are **absorbed**
- Output is **generated**, can **drive multiple inputs**

Quantum Computing

- **Memory** is made of **Qubits**, each holding a ***probability density function*** over 0 and 1
  - Qubit value collapses to 0 or 1 when read
  - Values have a limited lifespan (decoherence)
- **Processor** (really PIM: processors in memory)
  - Gates: **`NOT`**, **`CNOT`**, **`CCNOT`**, **`SWAP`**, **`CSWAP`**…
  - Often need **ancilla** (aka, garbage) qubits
  - **Fanout** is **not** allowed

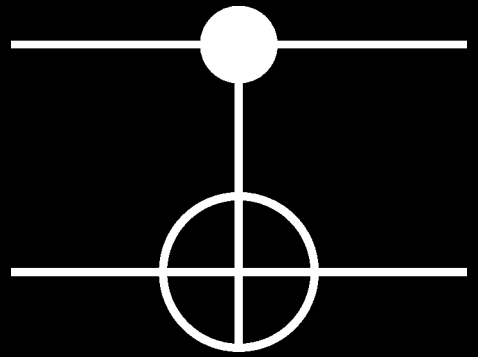
Quantum Gate Types: Pauli `x`

$$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

- **Pauli `X`** is also known as **`NOT`**
  - Rotates Bloch Sphere around X by π radians
  - Functions like conventional **`NOT`**
  - **`NOT`** is its own inverse

Quantum Gate Types: CNOT

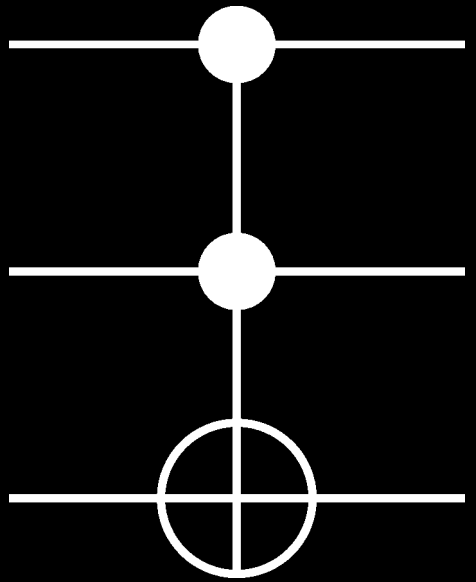
$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$



- **CNOT** is the **Controlled NOT** gate
  - Top input is control, passes thru unchanged
  - Bottom input is inverted where control is 1
  - Both inputs can’t be the same Qubit
  - Similar to conventional **XOR** gate

Quantum Gate Types: Toffoli

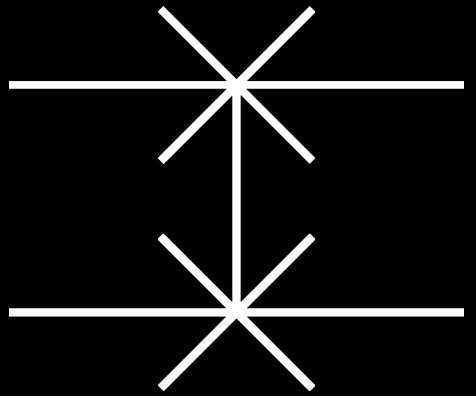
$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$



- **Toffoli** is also known as **CCNOT**, **Controlled Controlled NOT**
  - A classical universal gate
  - Control inputs pass unchanged
  - Inverts where both control inputs are 1
  - Behaves like `C = (A AND B) XOR C`

Quantum Gate Types: SWAP

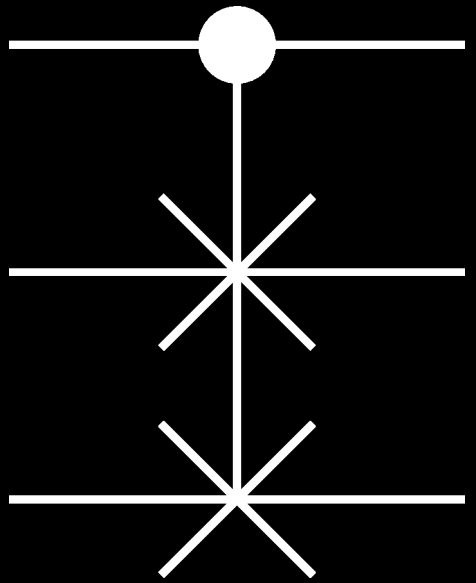
$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



- **SWAP** exchanges values of two Qubits
  - Seems pointless...
    but this is a **reversible assignment**

Quantum Gate Types: Fredkin

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$



- **Fredkin** is also known as **`CSWAP`**, **Controlled `SWAP`**
  - A classical universal gate… and *billiard-ball* conservative
  - Control passes unchanged
  - Inputs are swapped where control is 1

Quantum Gate Types: Hadamard

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

- **Hadamard** is not like any conventional gate
  - A qubit can only be initialized to 0 or 1
  - Hadamard operator converts that into the equiprobable superposed state

Quantum Gate Types: Hadamard

- If *applied in parallel* to ***E*** Qubits, the result is the equiprobable ***E***-way entangled superposition
- Two-qubit **Hadamard** is:

$$H_2 = H \otimes H = \frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix} \otimes \frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix} = \frac{1}{2}\begin{bmatrix}1 & 1 & 1 & 1\\ 1 & -1 & 1 & -1\\ 1 & 1 & -1 & -1\\ 1 & -1 & -1 & 1\end{bmatrix}$$

- Applying to two 0 qubits:

$$H_2|00\rangle = \frac{1}{2}\begin{bmatrix}1 & 1 & 1 & 1\\ 1 & -1 & 1 & -1\\ 1 & 1 & -1 & -1\\ 1 & -1 & -1 & 1\end{bmatrix}\begin{bmatrix}1\\ 0\\ 0\\ 0\end{bmatrix} = \frac{1}{2}\begin{bmatrix}1\\ 1\\ 1\\ 1\end{bmatrix} = \frac{1}{2}|00\rangle + \frac{1}{2}|01\rangle + \frac{1}{2}|10\rangle + \frac{1}{2}|11\rangle = \frac{|00\rangle + |01\rangle + |10\rangle + |11\rangle}{2}$$

Quantum Gate Types: Pauli `y`

$$\begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$$

- **Pauli `y`** is like combined **`x`** and **`z`**
  - Rotates Bloch Sphere around Y by π radians
  - **`y`** is its own inverse

Quantum Gate Types: Pauli z

$$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

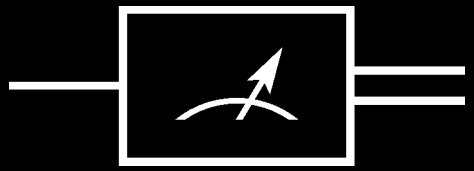
- **Pauli z** flips phase sign
  - Rotates Bloch Sphere around Z by π radians
  - Doesn't alter measured value
  - **z** is its own inverse

Quantum Gate Types: Phase

$$\begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}$$

- Phase shift **`S`** by ***i***
- There is also a **`P(ψ)`** gate that shifts by $e^{i\varphi}$
- **`P(π/4)`** is also known as **`T`**
- There are parametric rotations about X, Y, Z
- The accuracy of **`ψ`** , etc. are unspecified...

Quantum Gate Types: Measurement



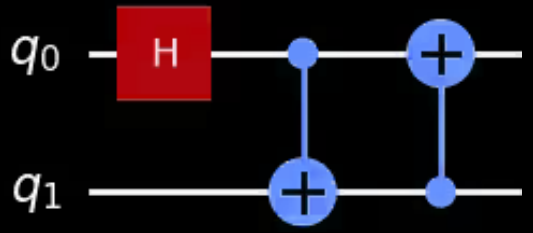
- **Measurement** collapses a superposition
  - Superposed Qubit becomes either 0 or 1
  - Superposed probability density function is randomly sampled, determines odds of 0 vs. 1

Exponentially cheap parallel computation…
but you can only measure one answer per run

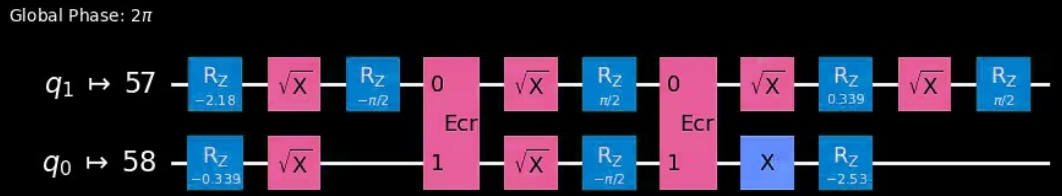
Sampling Without Measuring?

- Use phase operations to control/select what will be measured
  - Phase kickback
  - **Grover's algorithm**
  - **Quantum Fourier Transform** (**QFT**)

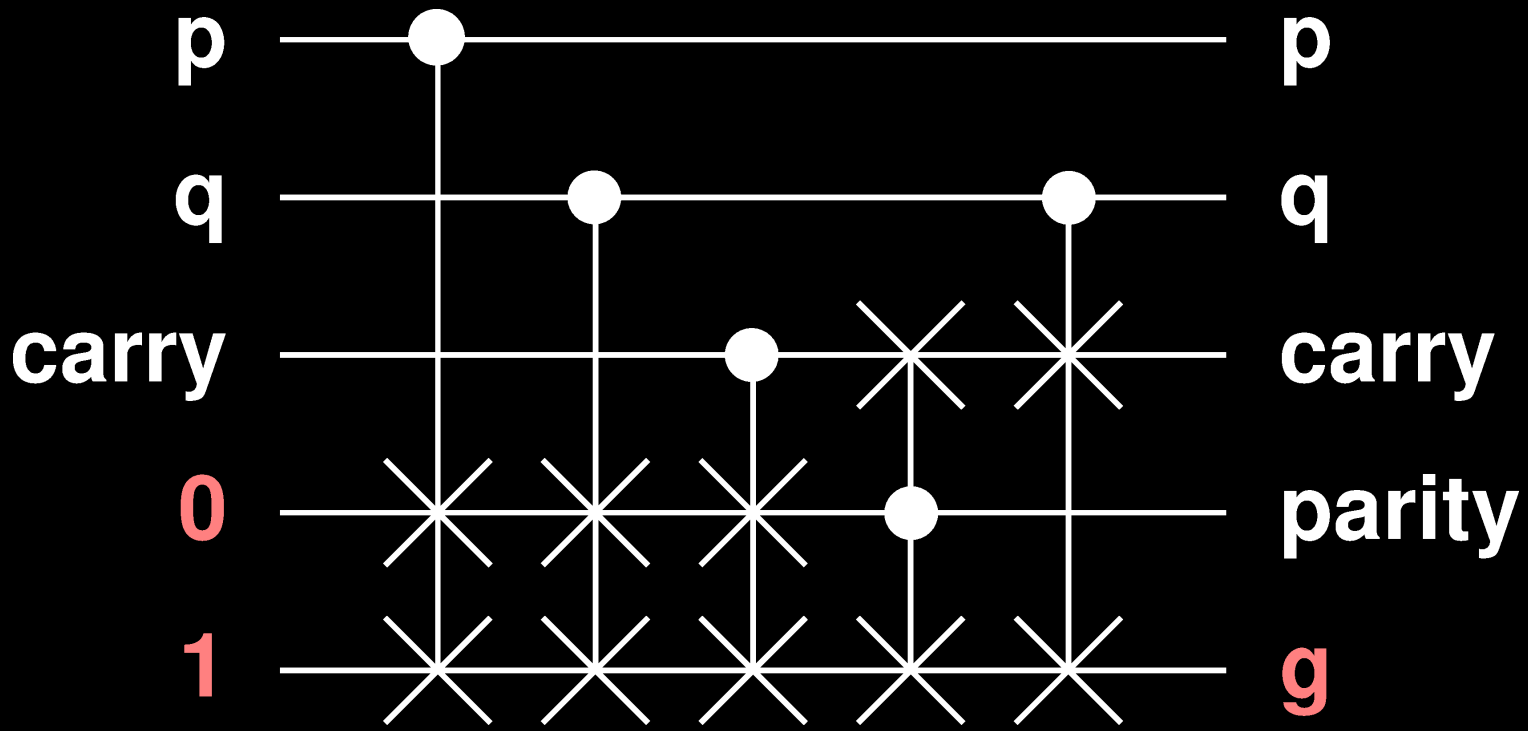
Which Gates Do You have?



- **Transpilation** is compilation
  - Translate to supported primitives
  - Allocate specific qubits
  - Apply optimization and scheduling

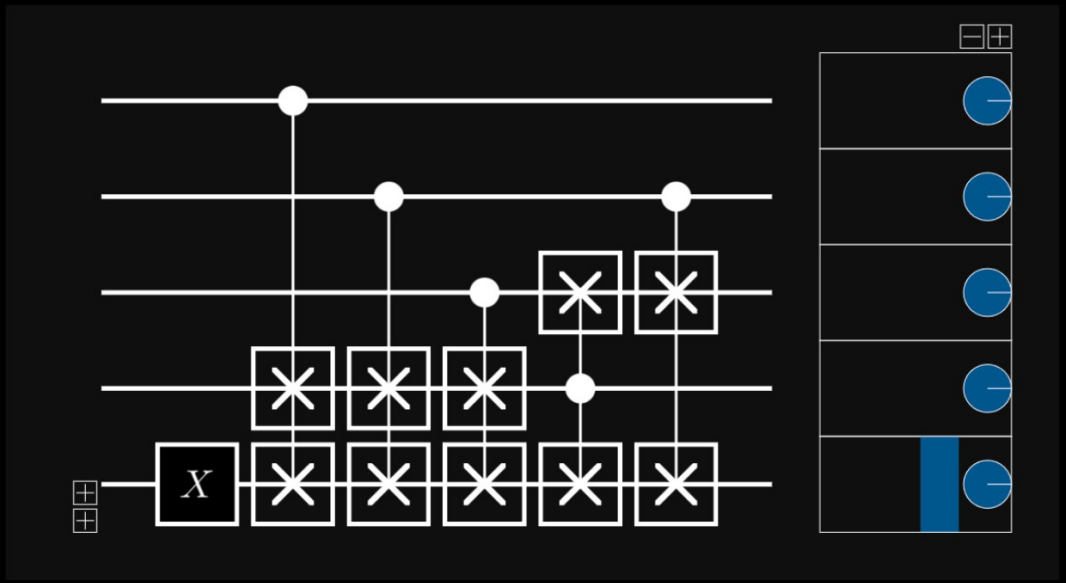


Let’s Build A 1-Bit Full Adder



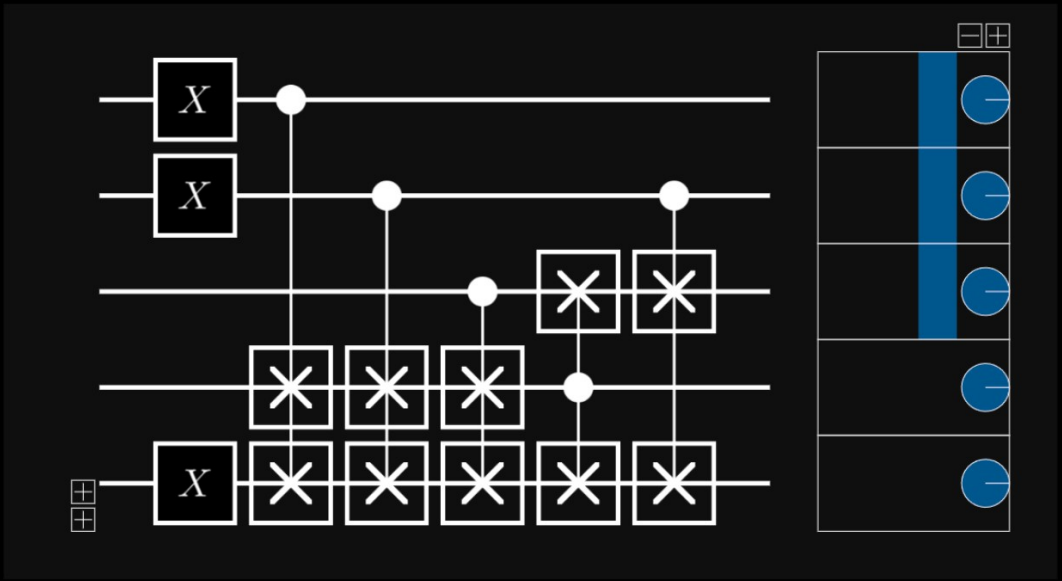
{carry, parity} = p + q + carry

Let’s Build A 1-Bit Full Adder



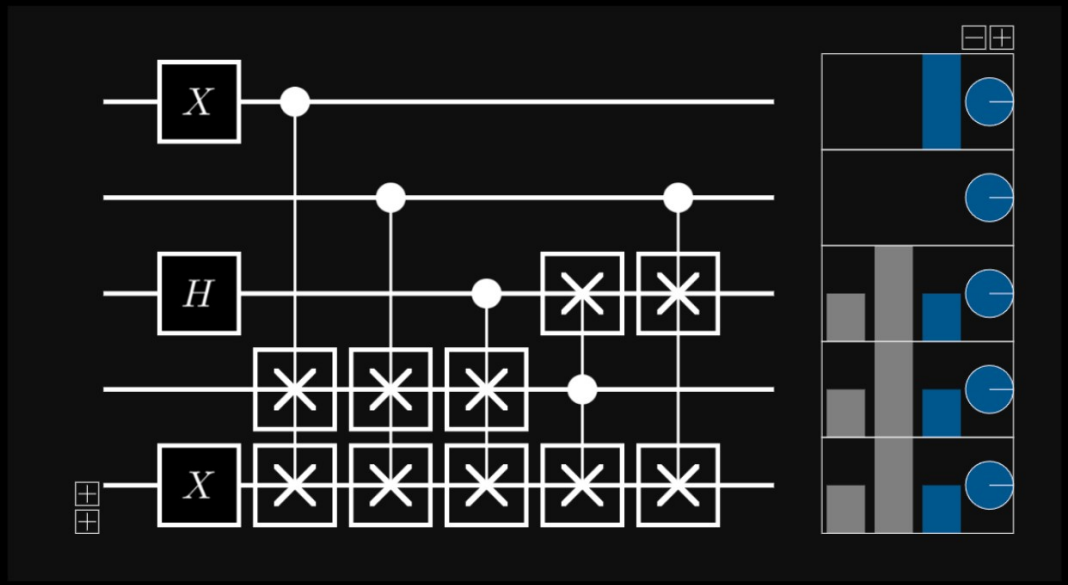
- Simulation using **MuqcsCraft**

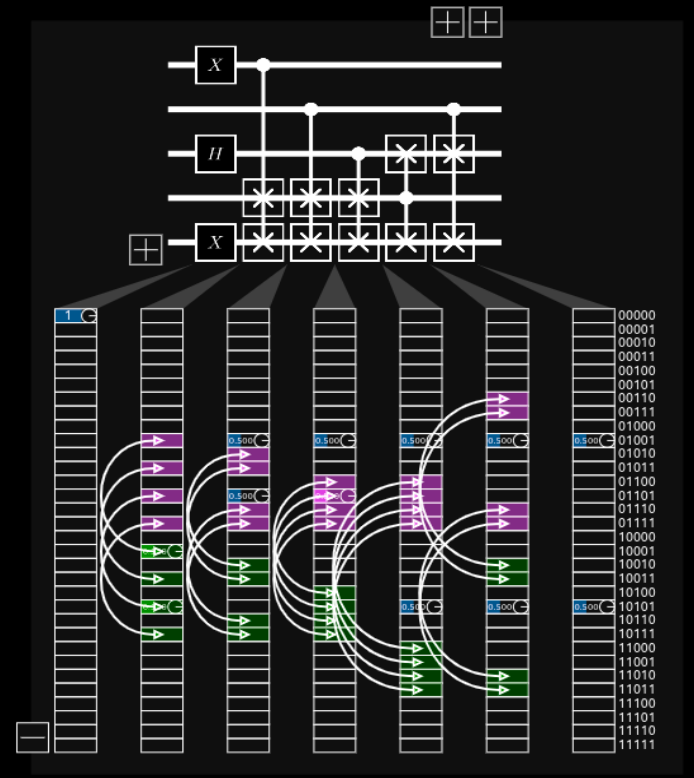
1+1+0



- Simulation using **MuqcsCraft**

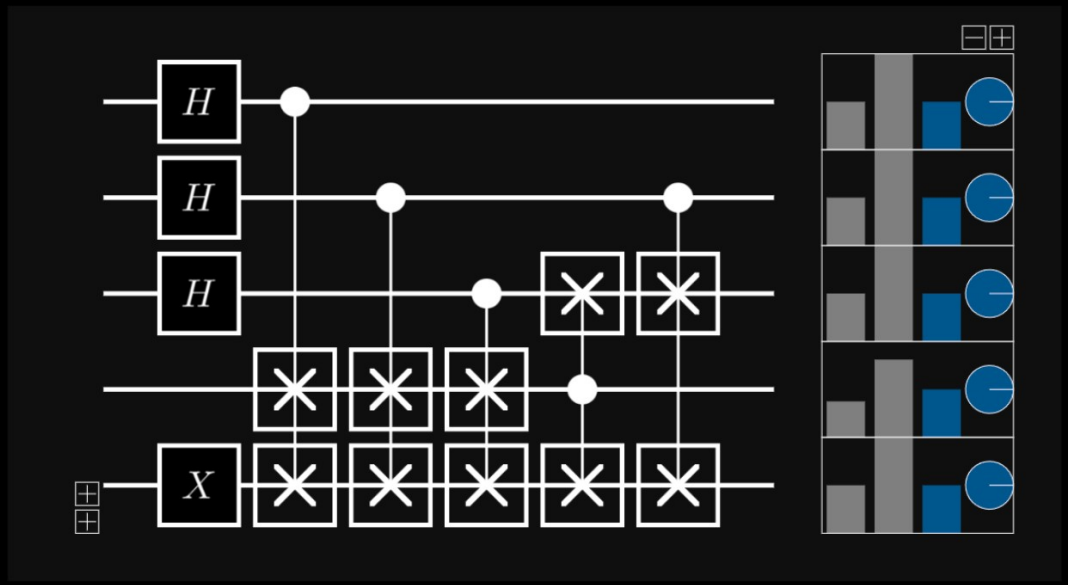
1+0+H

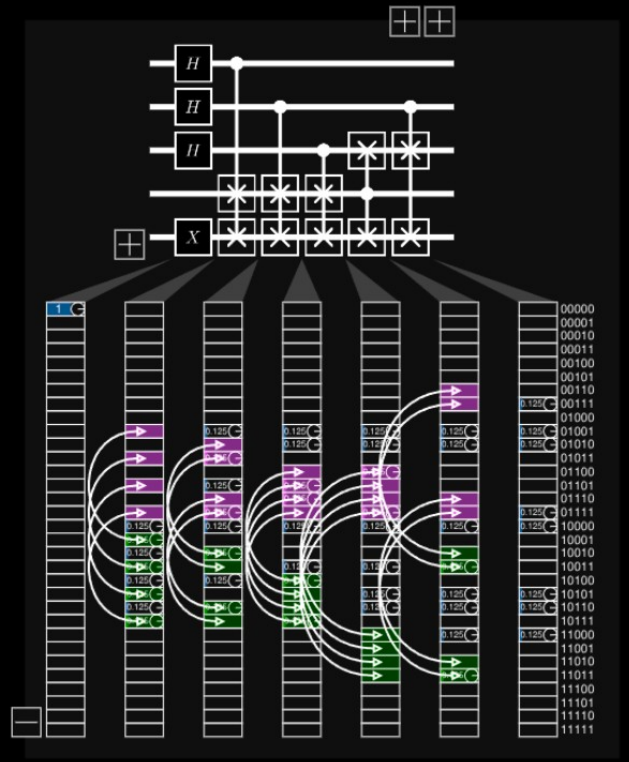




- Simulation using **MuqcsCraft**

H+H+H





- Simulation using **MuqcsCraft**

Quantum Supremacy or Advantage

Solving a useful problem faster than any classical computer could

- 2019 Google's 53-qubit **Sycamore**
- 2020 China's 113-qubit **Jiuzhang**
- 2021 IBM's 127-qubit **Eagle**
- 2024 Google's 105-qubit **Willow ...**

Is **Random Circuit Sampling** useful?

Shor’s Algorithm

*The **Quantum Algorithm** everyone hears about*

- Fastest algorithm to factor large semiprimes
- Most encryption based on it being hard to factor a large semiprime
- Shor’s Algorithm has **Quantum Subroutines**, but it is mostly run on a conventional computer

What Shor's Algorithm Actually Does

1. Pick random $a \in \{2, \ldots, N-1\}$
2. Let d = **gcd(a,N)**
3. If $d \geq 2$, d is a factor and we're done
4. Let r = **order of a % N** (this uses QFT)
5. If r is odd, fail
6. Let d = **$\gcd(a^{r/2} - 1, N)$**
7. If $d \geq 2$, d is a factor; else fail

Is Quantum Computing THE Answer?

So, What Is Quantum Good For?

- Problems where:
  - You need to try all possible values (quantum algorithms change logic, not values)
  - You need just one or some answers, not all
  - You don’t mind occasionally wrong answers
  - Combinatorial logic operating on few qubits

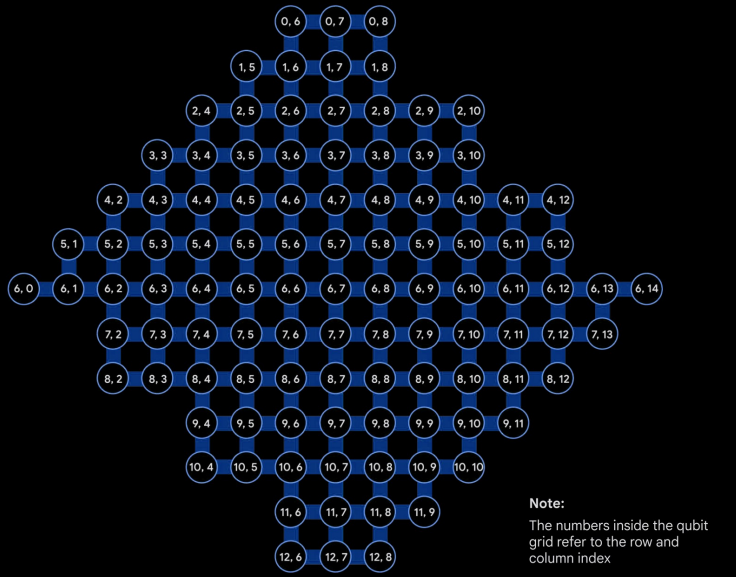
- **Quantum computers are special-purpose attached accelerators**, sort-of like GPUs

So, What Do We Have Now?

- NISQ: Noisy Intermediate-Scale Quantum
  - Tens to a few thousand qubits
  - Short coherence time
  - Noisy, with gate and measurement errors
- Work arounds
  - Quantum approximately solves subproblems
  - Error correction is a hot research area!

For example: Google Willow

- Willow “connectivity” with 105 qubits is 3.47 average, 5 max
- Gate(q): 0.036% error
  Gate(q,q): 0.14%
  Measurement: 0.67%
- Coherent for ~98μs, 40 gates



Conclusions

- Quantum computing is a way past Moore's Law, for very specific types of computations
- Quantum computers still have a long way to go
  - Quantum hardware *might* never get there
  - Thinking about quantum algorithms often yields faster algorithms for conventional computers
- RSA cryptography uses 2048-bit keys, **Shor's Algorithm** would need lots more qubits to break it

http://QERKY.ORG

Quantum Education and Research in KY

Ready for a short quiz?

Is this a Quantum Computer?

Is this a Quantum Computer?

Yup!

Google
Sycamore

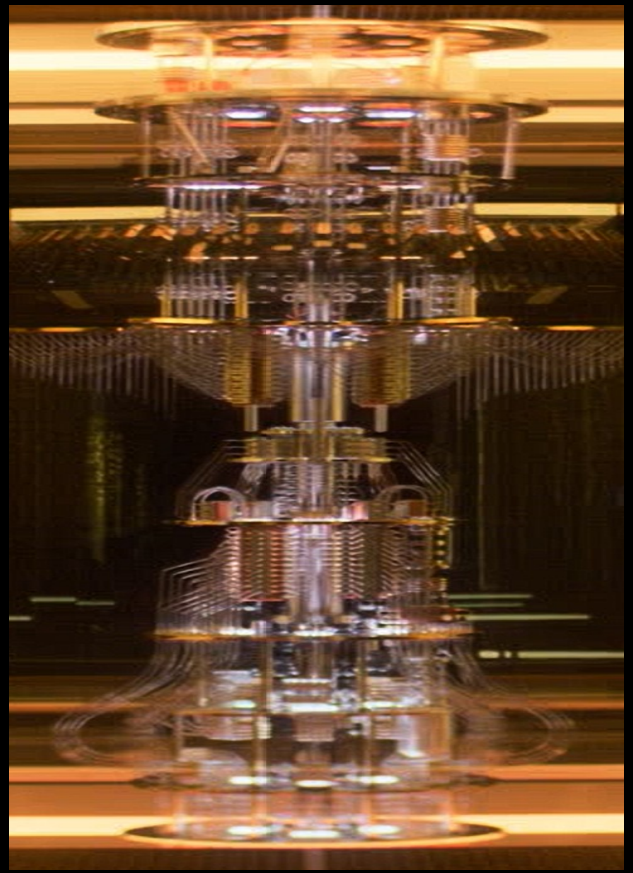
Is this a Quantum Computer?

Is this a Quantum Computer?

Yup!

SpinQ Gemini Mini
~$8K

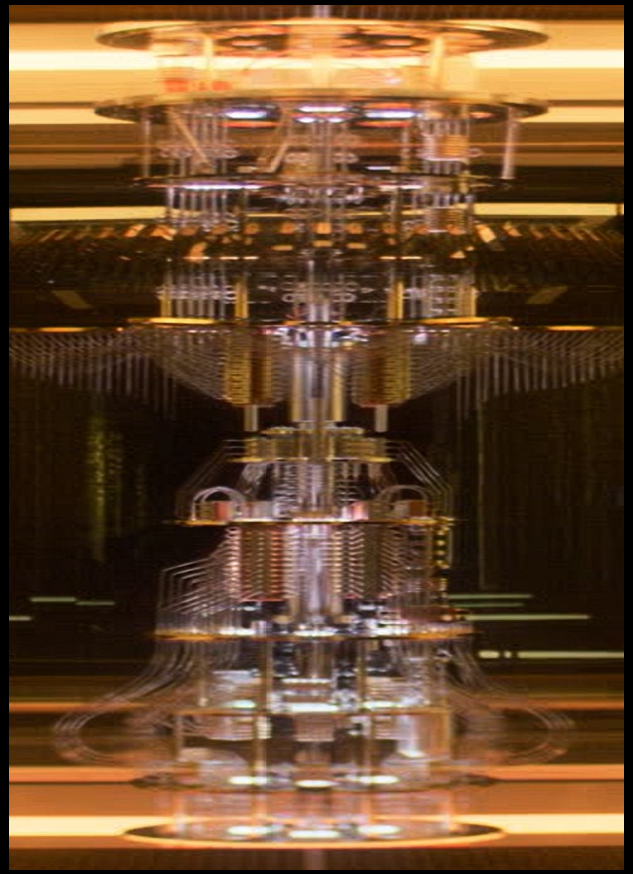
Is this a Quantum Computer?



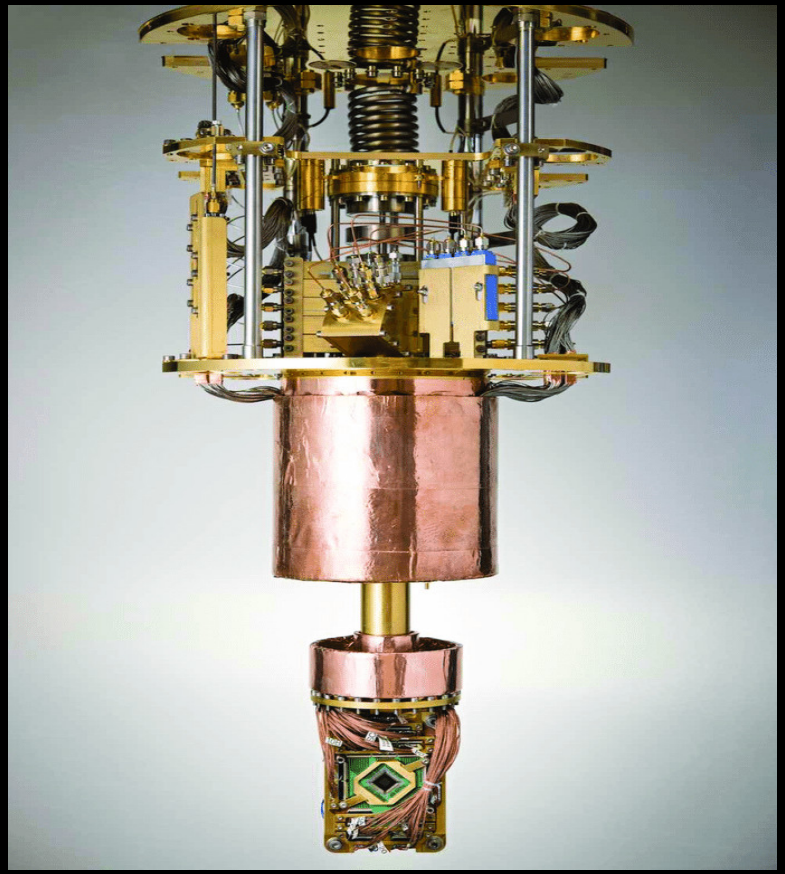
Is this a Quantum Computer?

Nope!

It is from the 2020 TV mini series **DEVS**



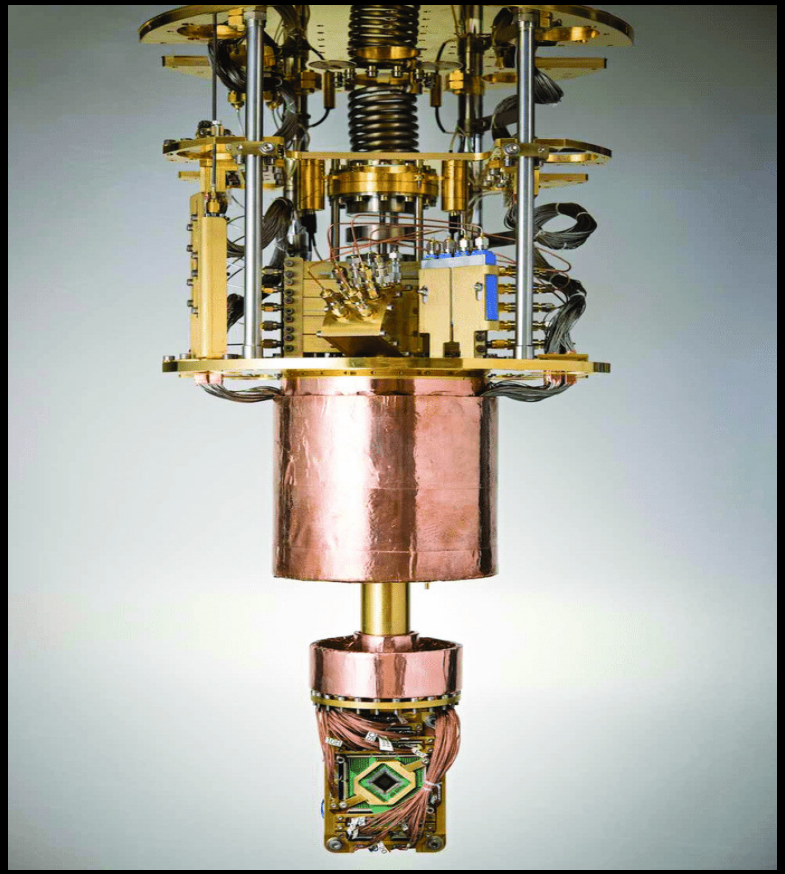
Is this a Quantum Computer?



Is this a Quantum Computer?

Yup!

It is a
D-Wave 2000Q



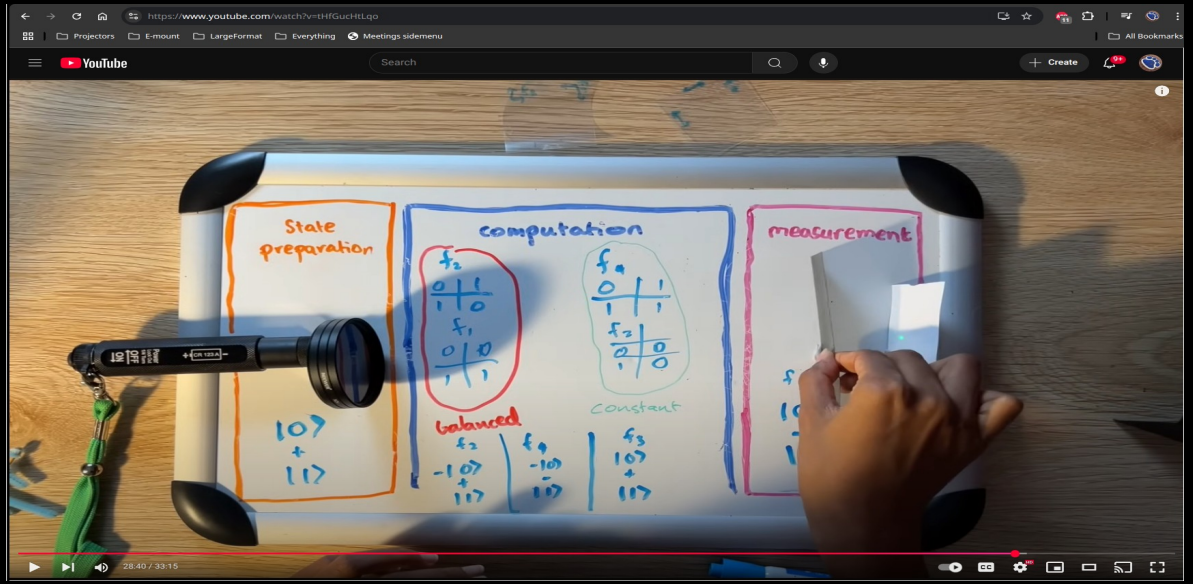
Is this a Quantum Computer?

Is this a Quantum Computer?

Not quite.

1/2-scale model of **Fujitsu**

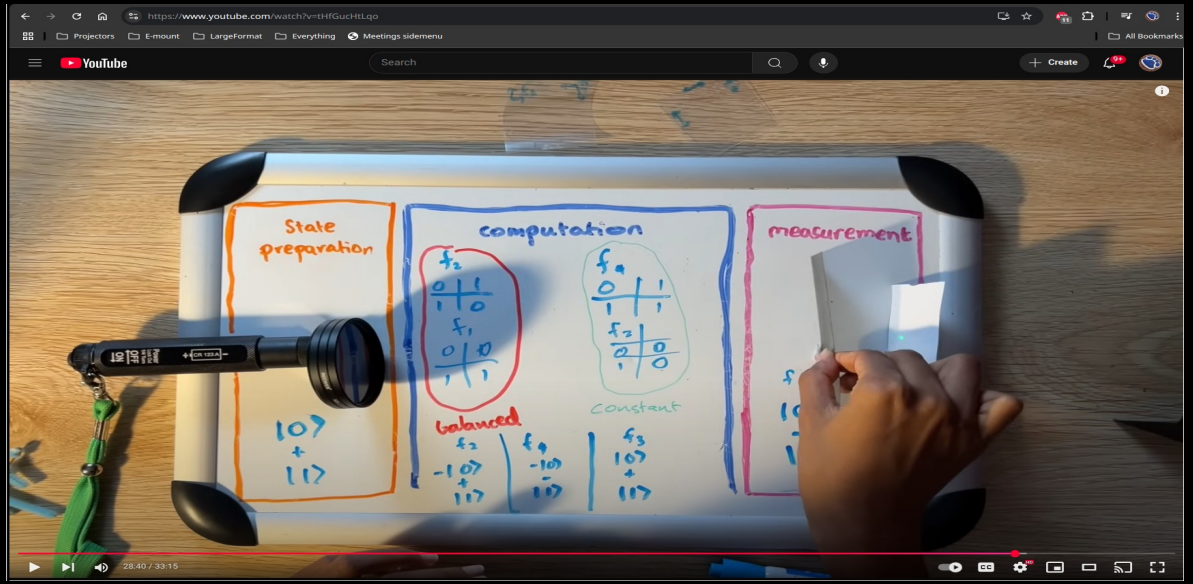
Is this a Quantum Computer?



Is this a Quantum Computer?

Yup.

A photonic qubit using polarization



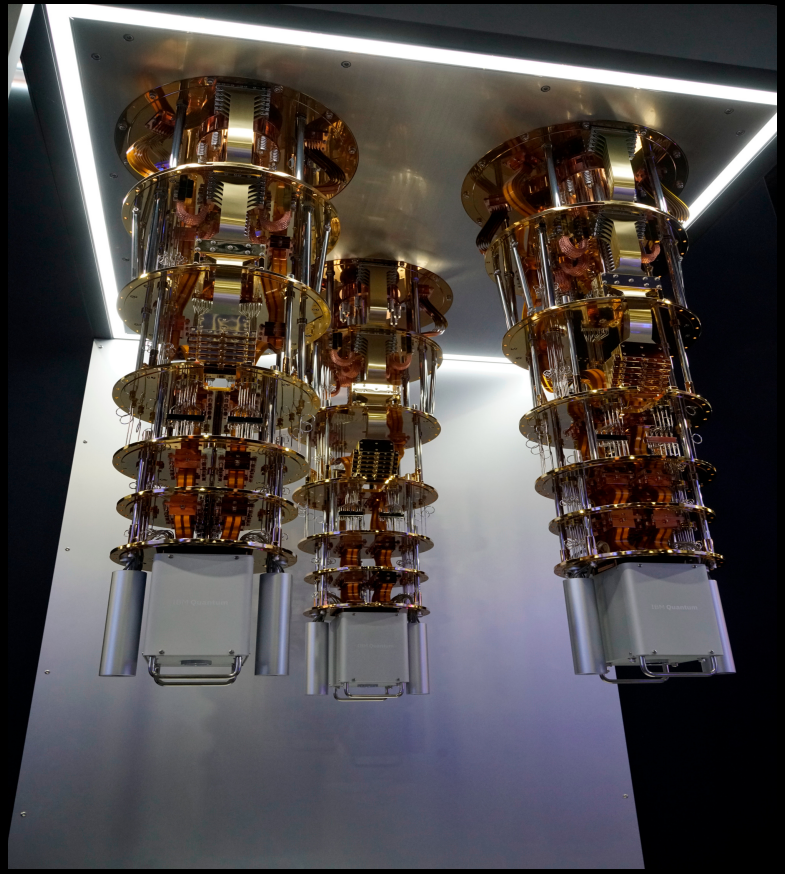
Is this a Quantum Computer?

Is this a Quantum Computer?

Yup!

IBM Q

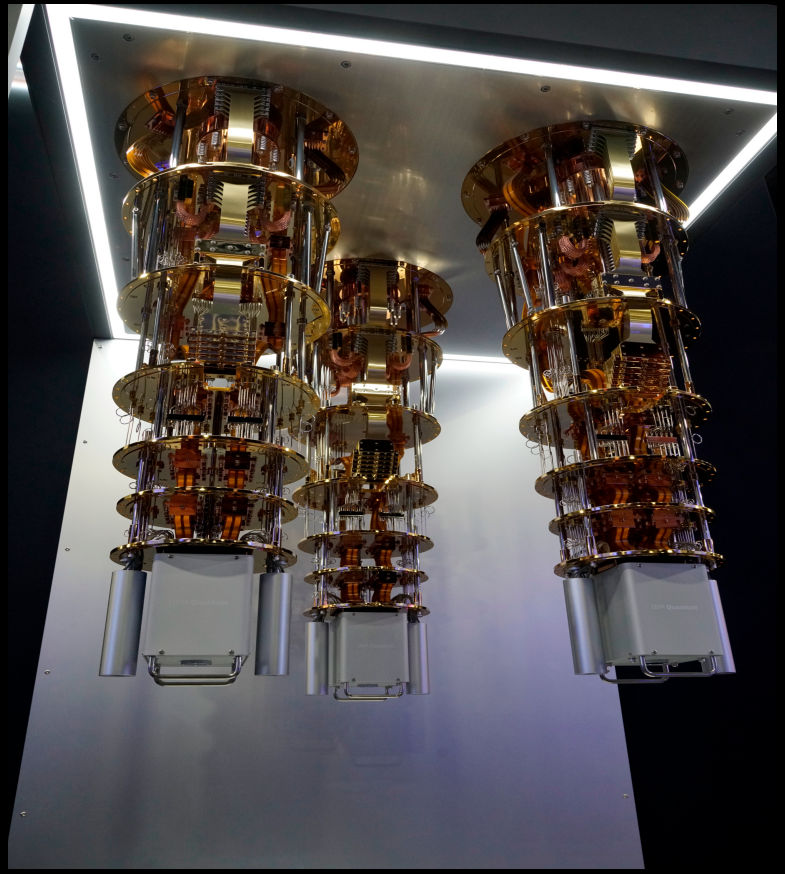
Is this 3 Quantum Computers?



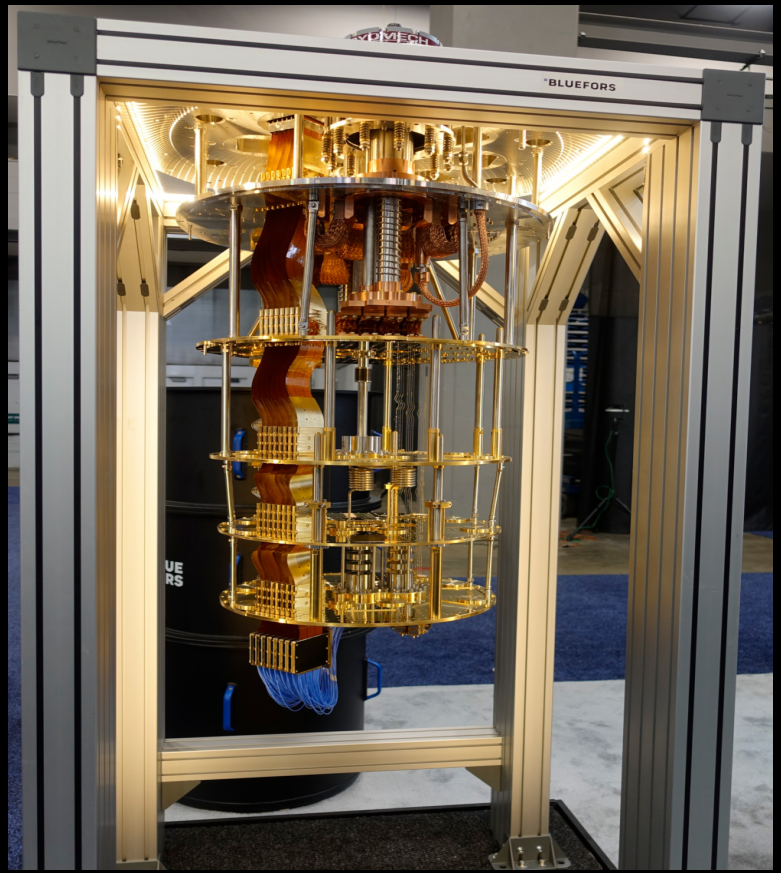
Is this 3 Quantum Computers?

Not quite.

80% scale
model of
IBM Quantum
System Two



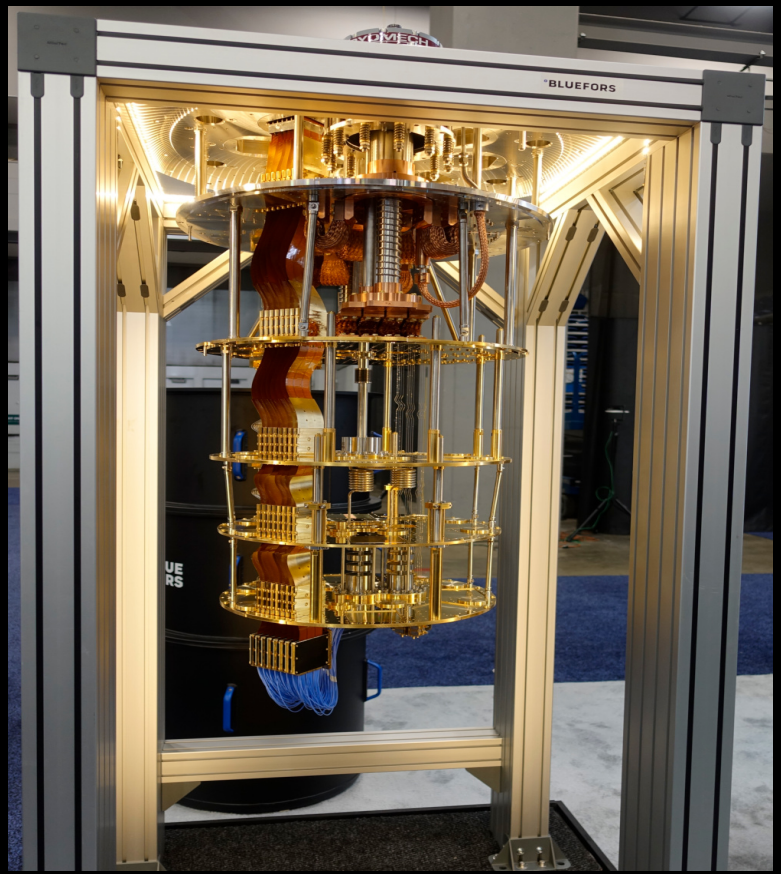
Is this a Quantum Computer?



Is this a Quantum Computer?

Nope!

BLUEFORS
This is a dummy; they make cooling, not computers



Is This A Quantum Computer?

Is This A Quantum Computer?

Nope.

PBP

Not quantum, but similar properties