THINK INSIDE THE BOX

QUANTUM COMPUTING WITH CAT QUBITS

An Introduction to Useful Quantum Computing by Alice & Bob

ALICE & BOB

WRITTEN BY

Niccolò Coppola, Raphaël Lescanne, Laurent Prost, Matija Žesko

CURATED BY

Brian Siegelwax

ILLUSTRATED BY

Guillaume Gennet

AUDIO VERSION NARRATED BY

Christopher Bishop

Listen to “Think Inside The Box” on:

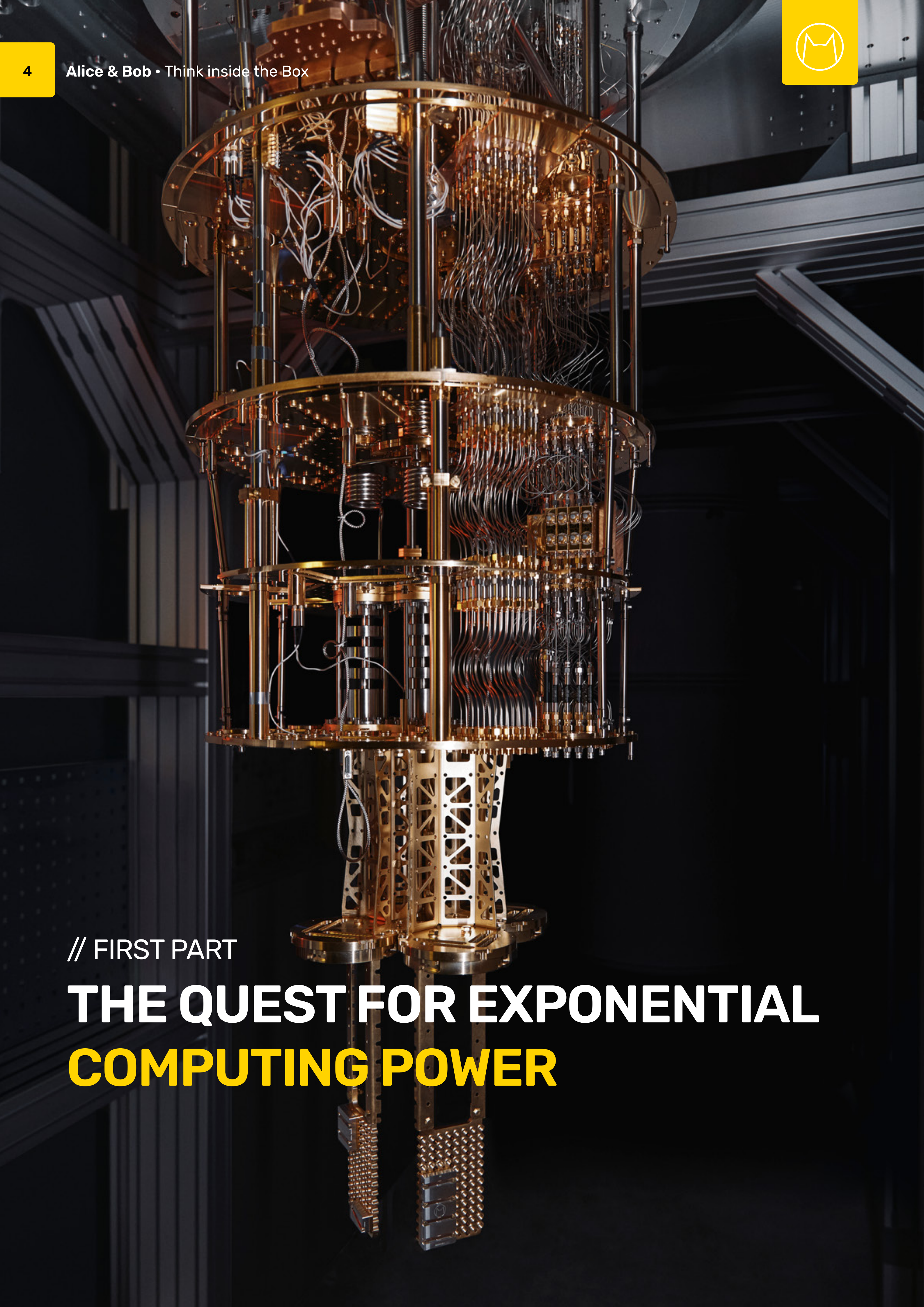
SPOTIFY

APPLE PODCASTS

To know more visit Alice & Bob’s website

www.alice-bob.com

CONTENTS



// FIRST PART

THE QUEST FOR EXPONENTIAL COMPUTING POWER

The Genesis of Quantum Computing: From Feynman's Vision to Shor's Algorithm

The year was 1981. The location was Pasadena, California. Richard Feynman had been working for over two decades as a physics professor at Caltech. In his youth, Feynman had been recruited by Robert Oppenheimer to run the computing unit at Los Alamos. So, in the early 1980s, with the modern computing revolution about to start, it should come as no surprise that he was already applying computers to cutting-edge physics research.

He hit a roadblock.

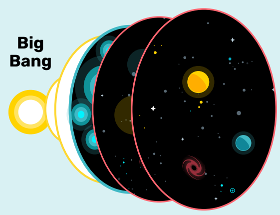
Feynman was trying to accurately simulate the behavior of large groups of quantum objects, such as systems of atoms and electrons. Fast forward to today, and our most powerful supercomputers can precisely simulate systems of only about fifty quantum objects. As you may imagine, more than four decades ago, Feynman couldn't simulate systems of more than a handful. He famously concluded:

> **"NATURE ISN'T CLASSICAL, DAMMIT, AND IF YOU WANT TO MAKE A SIMULATION OF NATURE, YOU'D BETTER MAKE IT QUANTUM MECHANICAL, AND BY GOLLY IT'S A WONDERFUL PROBLEM, BECAUSE IT DOESN'T LOOK SO EASY."**

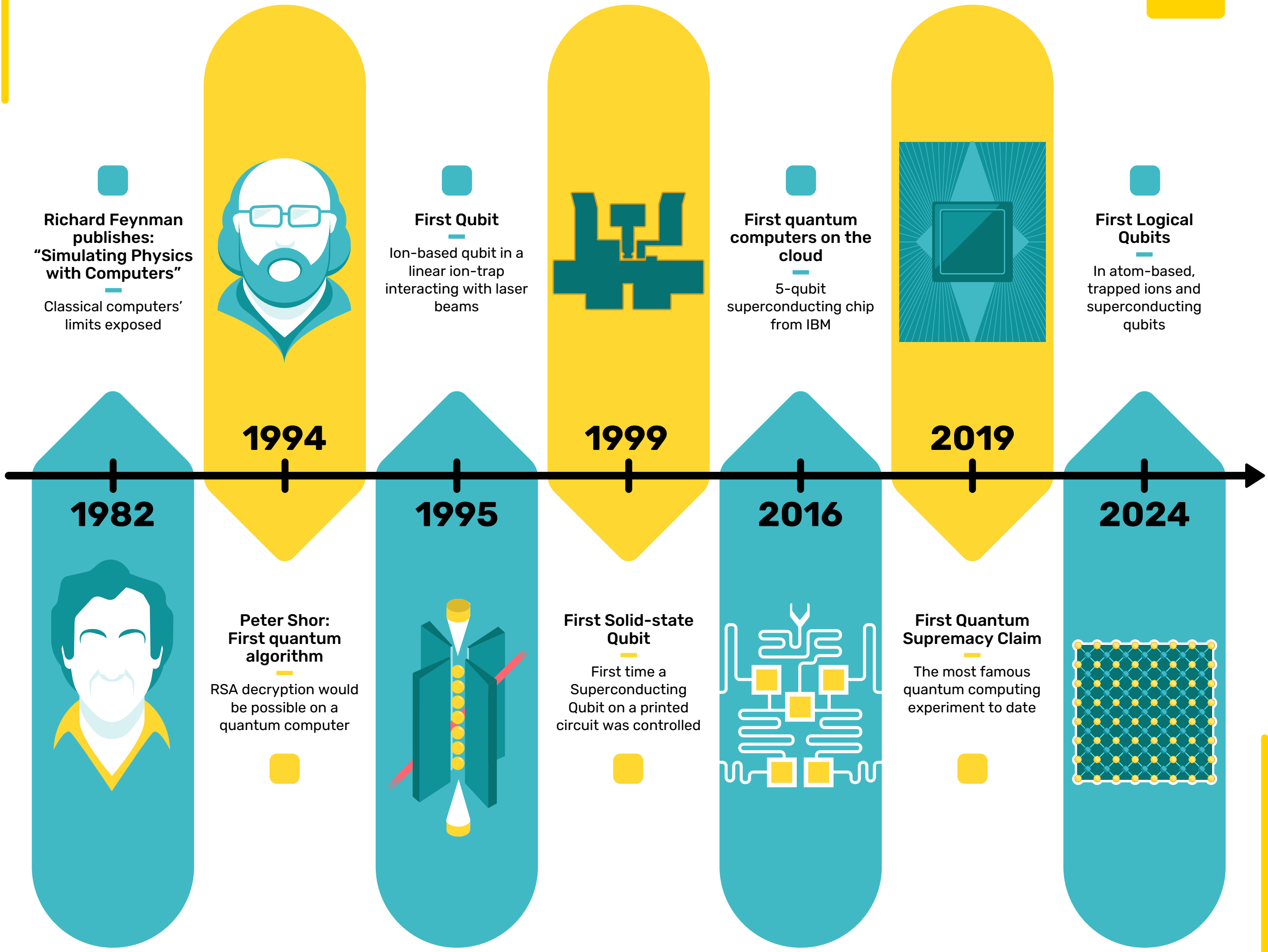
With this legendary statement, Feynman popularized the idea of a 'quantum computer.'

A decade later, Peter Shor, an American mathematician then at Bell Labs, devised one of the first quantum algorithms. Multiplying two numbers is easy for a classical (non-quantum) computer, but performing the inverse operation, factoring a number, is very hard. As numbers become larger, in fact, this problem quickly becomes intractable. This problem is so difficult that modern data encryption exploits this intractability to protect our information. Unfortunately, by leveraging the properties of quantum mechanics, Shor found a quantum algorithm that dramatically accelerates the solution to this inverse problem. This discovery puts today's data security at risk once we create a quantum computer powerful enough to run it.

WHAT IT TAKES TO FACTOR A NUMBER

Key to decrypt (bits)	**512**	×2 → **1024**	×2 → **2048** (Today's RSA encryption)
CLASSICAL COMPUTER HP Frontier Supercomputer	 **0.5 Milliseconds** (20x faster than a hummingbird flaps its wings)	 **3 Months**	 **6 Trillion years** (400x The age of the universe)
QUANTUM COMPUTER Alice & Bob's Architecture	**10 Hours**	**1 Day**	**1 Week**

Today's fastest classical computer would take four-hundred times the age of the universe to factor a 2048-bit number. Hopefully, we can all agree that this is impractical. This estimation perfectly illustrates, though, why this mathematical problem is used to encrypt most of our information, ranging from your email password to state secrets. In stark contrast, a quantum computer running Shor's algorithm could crack RSA-2048 in just a matter of days.



A TIMELINE OF QUANTUM COMPUTING

Feynman realized that only a quantum computer could effectively simulate the fundamental physics of large quantum systems. Shor demonstrated that quantum computing could be a tool for much more. It might impact the entire field of computer science.

PETER SHOR'S DISCOVERY UNOFFICIALLY STARTED THE RACE TO BUILD A QUANTUM COMPUTER.

Quantum computing remained a theoretical topic until 1995, when the first ion-based qubits (quantum bits) appeared[1]. Then, in 1999, the first superconducting qubits were created at NEC laboratories[2]. By the start of the millennium, quantum computing was already catching the interest of the world's biggest tech companies. The first cloud-accessible quantum computer prototypes appeared in 2016, courtesy of IBM[3], and the first claim of 'quantum supremacy' was made in 2019 by Google[2]. Other tech heavyweights, such as Microsoft, Amazon, and Intel, soon joined the hardware race, as did many startups.

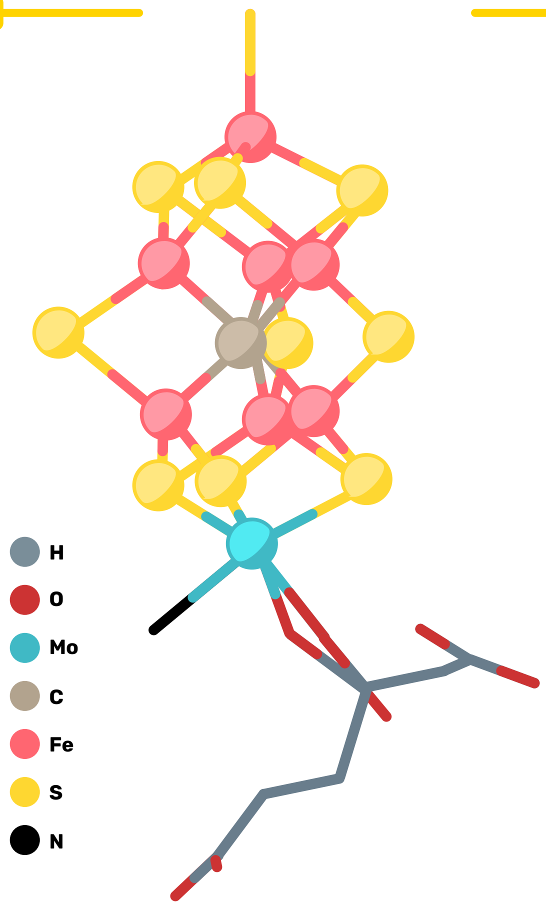
IN A NUTSHELL

Physicists have been dreaming about quantum computers since the 1980s, when Feynman realized that only these machines could simulate large quantum systems. However, the alarm clock didn't start ringing until the 1990s, when Peter Shor discovered his eponymous algorithm. This breakthrough awakened global tech companies to new possibilities, prompting them to start turning these dream machines into reality.

Beyond Classical Limits: The Promise of Quantum Computing

Richard Feynman's challenge was that the complexity of simulating large quantum systems, such as molecules, grows exponentially with the number of quantum objects (atoms and electrons) they contain:

- A handful of objects can be simulated with pen and paper.
- 20-30 objects can be simulated with an average laptop.
- 40-50 objects can be simulated with the best classical supercomputers.
- Simulating a system with more than ~50 objects is not possible with today's technology.



Around the world, fertilizer production relies on industrial nitrogen fixation using the extremely energy-intensive Haber-Bosch process. FeMoCo, illustrated above, is a complex metallocluster responsible for nitrogen fixation in plants. Impossible for supercomputers, a quantum computer with ~1,000 logical qubits could simulate its mechanisms. This could lead to more energy-efficient, biologically-inspired methods of artificial nitrogen fixation.

Exponential scaling means that tomorrow's supercomputers won't fare much better with large quantum systems than today's. However, as Richard Feynman noted, nature has no problem with this at all. Why is this so hard for us humans and our machines, then?

Our classical computing model decomposes information into fundamental units called 'bits.' This information is physically stored on massive quantities of physical systems, each with two possible states, which we label as 0 and 1. The computer then performs tasks by applying fundamental operations, called 'gates,' to these bits.

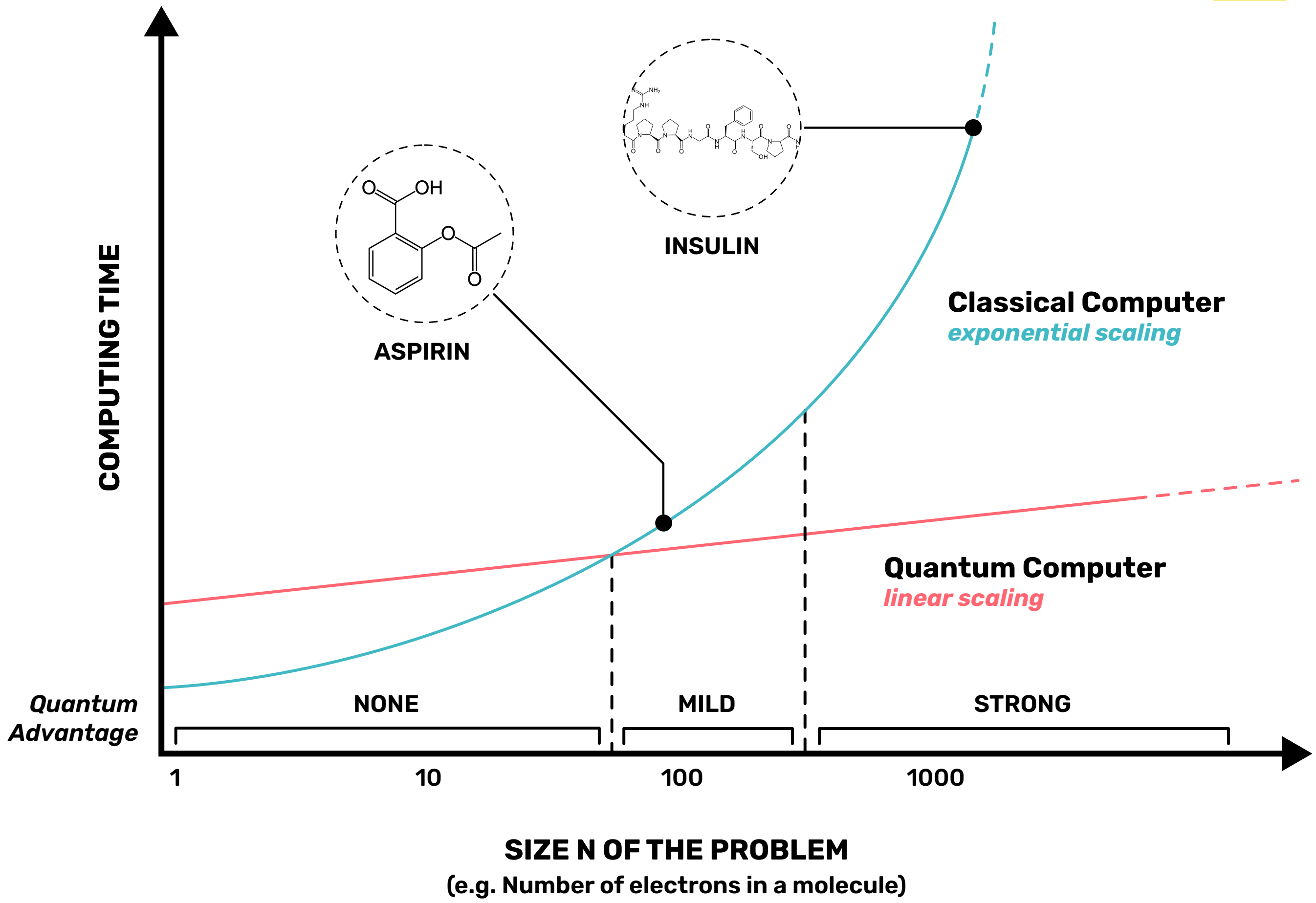
The problem we encounter is that the number of classical gates needed to simulate quantum systems or factor numbers can grow exponentially. At a minimum, this number grows 'super-polynomially' (faster than any polynomial) with the size of the problem.

Fortunately, we can overcome this problem by rethinking the way we approach computation. By using quantum bits and quantum gates, which are based on principles of quantum physics such as superposition and entanglement, we can remove the limitations of the classical computing model and create a new, richer quantum computing one.

THIS NEW QUANTUM MODEL EXTENDS THE CLASSICAL ONE BY STORING MORE INFORMATION IN EACH FUNDAMENTAL UNIT AND PERFORMING MORE COMPLEX TRANSFORMATIONS WITH EVERY GATE.

The shift to leveraging quantum physics allows us to significantly reduce the number of gates needed[5], enabling us to solve complex problems more efficiently. A quantum computer that can outperform our best classical computers in solving useful tasks would demonstrate so-called 'quantum advantage.'

QUANTUM ADVANTAGE DEPENDS ON THE PROBLEM SIZE



Quantum advantage depends on the specific problem being solved and its size. In this illustrative example of quantum simulation in the pharmaceutical industry, we see that as the target molecule increases in size, quantum computing becomes more advantageous due to its favorable computational scaling compared to classical computing. However, achieving quantum advantage also requires quantum computers to scale effectively.

Over the past three decades, researchers have discovered quantum algorithms that could offer speed-ups for a variety of use cases, including:

- Quantum Systems Simulation
- Linear Algebra
- Optimization
- Search Algorithms
- Machine Learning
- Factorization
- And many more.

A quantum computer that can solve industrially relevant problems with these algorithms would be revolutionary. It would enable breakthroughs in drug discovery, materials science, fundamental physics, chemistry, biology, and other fields.

How do we define 'revolutionary'? The economic value that full-fledged quantum computing could unlock was recently estimated between $850 billion[6] and $2 trillion [7].

IN A NUTSHELL

Classical computers hit a wall when simulating exactly systems larger than ~50 quantum objects, but quantum computers break this wall down by using qubits and quantum gates. From scientific research to various industry sectors, quantum computing could incite a trillion-dollar revolution.

Note to the CxO: In Search of Near-Term Value

Although this whitepaper dives deeply into building a useful general-purpose quantum computer, you might be interested to know that application-specific computers will likely start creating business value earlier. After all, this happened with classical computing. And since you're probably now wondering if your business might find near-term value, here are the key factors we evaluate:

1. Near-term feasibility: These are applications that can run on a fault-tolerant quantum computer with around 100 to 1,000 logical qubits.

2. Classical intractability: These are problems that take too much time or cost too much money to be economically viable with classical computing power.

3. Outsize value potential: In most cases, 'outsize' means an ROI of at least 100%. We're looking for scenarios where quantum computers can at least double your investment over three years.

When applying these filters to industries like chemicals or pharmaceuticals, molecular simulation emerges as one of the most promising use cases. The complex correlations in molecules are not well captured using classical methods, leading to unreliable simulations that then require expensive and time-consuming experiments.

If you're in the chemical sector, for example, we suggest looking at the catalysis market. It's worth around $30 billion and is expected to grow at a healthy 4.6% annual rate[8]. Capturing just a fraction of this market with quantum simulations could unlock significant value, and this is only one slice of this multi-trillion-dollar industry.

Similarly, in pharma, a DARPA-sponsored study estimates that improving molecular screening with low-field NMR spectra could save $65 million per year[9]. That might not seem like a huge number when discussing a trillion-dollar industry, but it's one link in a very long R&D chain. Once quantum computers start reducing the time to bring a drug to market, the gains could be substantial. Using the 26% net margin typical for drug manufacturers[10] as an example, accelerating the development of a single blockbuster drug (those generating over $1 billion annually) by just one year could add $260 million in profit.

OVERVIEW OF QUANTUM ALGORITHMS AND USE CASES

SHORTER TERM ⟵⟶ **LONGER TERM**

	Molecular simulation	Quantum optimization	Quantum Monte Carlo	Machine learning	HHL	Decryption[1]
Life sciences	Calculating a drug's binding affinity	Optimizing the location of clinical trial sites	Predicting the spread of disease in epidemics	Improving image classification in diagnostics	Modelling forces for protein-folding simulations	Protecting patient data privacy
Chemicals	Simulating the reaction pathway in synthesis	Optimizing the production process of chemicals	Simulating meso-scale reactor processes	Predicting the properties of new chemicals	Solving fluid dynamics in reaction vessels	Protecting data related to IP and trade secrets
Energy	Designing new materials for carbon capture	Optimizing power dispatching in an electric grid	Forecasting energy prices in the market	Predicting energy production from weather patterns	Solving DC power flow calculations in electrical grids	Protecting access to data on grid infrastructure
Telecom	Designing new semiconductor materials	Optimizing antenna placement	Stress-testing network resilience	Improving customer segmentation	Solving EM-field calculations in antenna design	Protecting the data exchanged over a network
Advanced manufacturing industries	Designing new batteries for electric vehicles	Optimizing the step sequence in car production	Improving the resilience of the supply chain	Improving fault detection in chip manufacturing	Solving aerodynamics simulations	Protecting communication connections
Logistics	N/A	Optimizing the route of a delivery service	Stress-testing logistic schedules for disruptions	Predicting maintenance needs in a fleet	Improving inventory management	Protecting personalized customer data
Finance	N/A	Optimizing the value of an asset portfolio	Modelling credit value at risk in capital allocation	Improving the detection of fraud in transactions	Estimating risk for the future value of an asset	Protecting customer transaction data

1. Decryption use cases imply using a quantum computer in penetration testing, e.g., by the company's IT security team.
Research conducted by Alice & Bob's quantum advisory team: The Box.

The Quantum Challenges: A Difficult Race Against Classical Computing

Classical computing has relied on the same fundamental physical building block, the transistor, since the 1950s. Transistors have proven to be fast, reliable, and scalable. They have stood the test of time, which is no small feat considering how much investment has been poured over the years into creating better alternatives. However, they have finally reached their physical limits in terms of miniaturization.

Furthermore, transistors are not suitable for use as qubits. This has forced the industry to seek out quantum analogs. Fortunately, since the first demonstration of physical qubit operations in 1995[11], we've witnessed strong progress in qubit hardware ranging from superconducting electrical circuits to arrays of trapped atoms. The largest quantum computers now have hundreds, if not thousands, of qubits, which is supposedly enough to demonstrate quantum advantage.

Unfortunately, this isn't happening. Claims of quantum advantage are usually squashed by computer scientists within weeks. This race has produced some tangible benefits, notably the development of quantum-inspired classical algorithms, yet practical quantum advantage continues to elude us.

What's up with that?

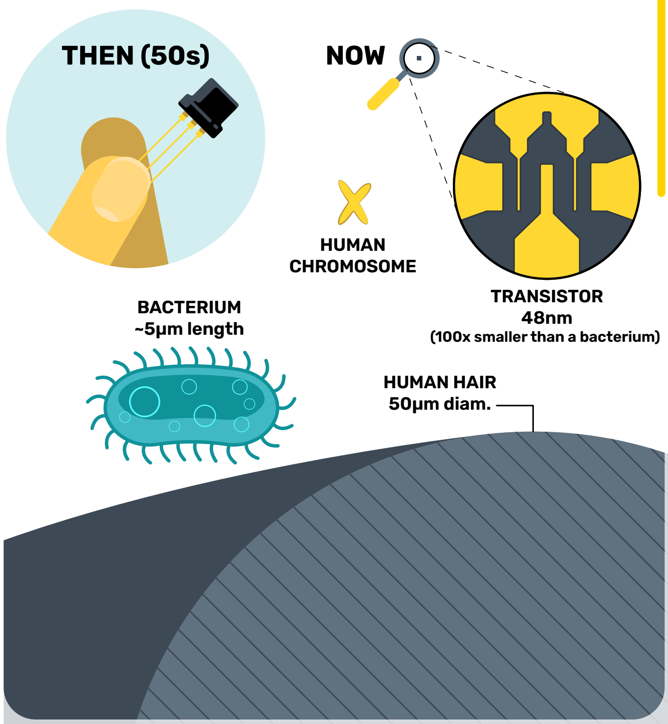
THE SIMPLE ANSWER IS THAT CURRENT QUANTUM COMPUTERS ARE UNRELIABLE.

Today's best hardware can barely execute a few thousand gates before the output becomes pure noise. Computation is corrupted by the interference between qubits and their environment, resulting in outputs that are indistinguishable from randomness.

THE REASON IS DECOHERENCE. A QUANTUM SYSTEM LEAKS INFORMATION ABOUT ITS STATE TO THE CLASSICAL WORLD SURROUNDING IT, AND IT STOPS BEHAVING 'QUANTUMLY.'

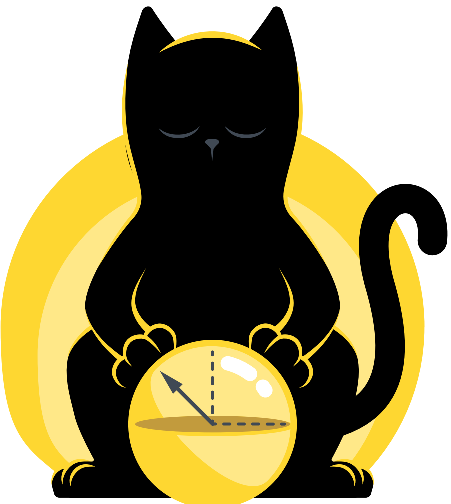
Significant progress has been made to protect quantum computers against decoherence, but we are still far from where we need to be.

THE TRANSISTOR



The first industrial-grade transistors, used in 1950s radios, were a few centimeters long. Modern transistors have been shrunk to the nanoscale, approaching their physical limits. The pace of miniaturization, known as Moore's Law, has significantly slowed down in recent years.

We at Alice & Bob have strongly believed from the beginning that quantum computers can only be useful if they escape decoherence.



DECOHERENCE IN QUANTUM COMPUTING IS ANALOGOUS TO GRAVITY IN SPACE EXPLORATION. IN EITHER SCENARIO, WE MUST ESCAPE SOME FUNDAMENTAL FORCE TO REACH OUR DESTINATION.

More than four years after our inception, our theoretical and experimental compasses show that we're heading in the right direction.

Today, the quantum industry as a whole is following the same path we believed in from the beginning.

We deeply respect the empirical attempts to squeeze utility out of currently available Noisy Intermediate-Scale Quantum (NISQ) computers. This has led to significant advancements in hardware quality and algorithm design. However, at Alice & Bob, we choose to bypass this approach.

WE'RE FOCUSED ON CREATING ERROR-CORRECTED COMPUTERS, CALLED FAULT-TOLERANT QUANTUM COMPUTERS (FTQC).

Only fault tolerance can lead to useful machines that demonstrate practical quantum advantage.

IN A NUTSHELL

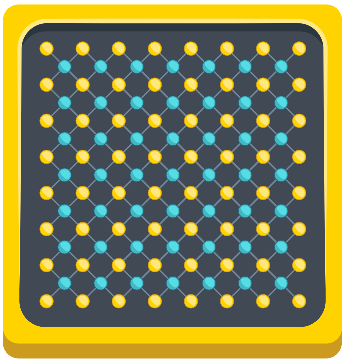
Surpassing the reliability and scalability of transistors has proven incredibly difficult. While many are attempting to build 'useful' quantum computers with Noisy Intermediate-Scale Quantum (NISQ) devices, we at Alice & Bob reject this approach. Instead, we're focused on building fault-tolerant quantum computers that can overcome decoherence, which is key to achieving practical quantum advantage.

The Three Steps Toward Useful Quantum Computers

A 'useful,' fault tolerant, quantum computer would solve classically intractable problems in relatively short timeframes, but the quantum circuits for this would require at least 100 error-corrected qubits and at least 100 million gates[12].

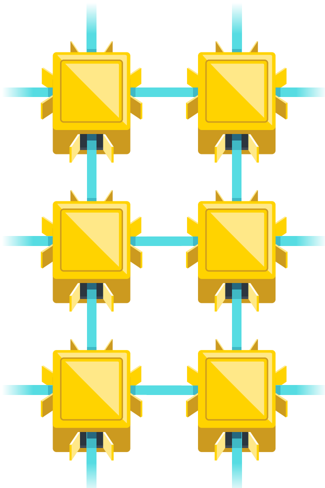
THE ROAD TO A MACHINE CAPABLE OF RUNNING SUCH CIRCUITS IS A LONG ONE. HOWEVER, IT CAN BE DIVIDED INTO THREE MAJOR INDUSTRY-WIDE STEPS:

1



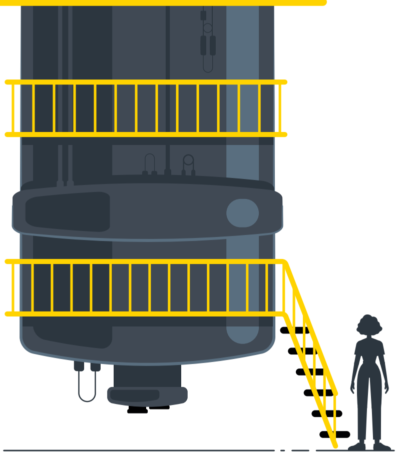
FIRST STEP: Fault-Tolerance
Control error-corrected qubits, a so-called 'logical qubits,' and logical gates capable of escaping decoherence.

2



SECOND STEP: Universality
Implement a universal set of error-corrected gates between these logical qubits, enabling any type of algorithm to run.

3



THIRD STEP: Scale
Have quantitatively and qualitatively enough logical qubits to achieve quantum advantage.

The first of these milestones is the most important. Fortunately, tremendous progress has been made over the last two years. We are living through a crucial moment in the history of quantum computing: the launch of the first logical qubits. Like Sputnik, the first satellite to escape Earth's gravity, these logical qubits escape decoherence but don't do anything particularly useful. Nonetheless, they are baby steps necessary to building universal quantum computers at scale.

This whitepaper will focus primarily on the first milestone and Alice & Bob's plan to achieve it, but it's important to note that these three steps are co-dependent in our roadmap. Our hardware is already incorporating elements of universality while pursuing fault-tolerance. The challenge of scaling up is being tackled by many players at various levels in the value chain. Examples include the nanofabrication of larger, denser chips, the engineering of more capacitive cryogenic systems, and the miniaturization of cabling.

We plan to go beyond creating logical qubits to building powerful, fault-tolerant quantum computers at scale. We describe our roadmap to get there in the fourth and last part of this whitepaper.

IN A NUTSHELL

A useful quantum computer must be resilient to errors while running circuits with millions of gates across more than 100 qubits. This journey has three key milestones: fault tolerance, universality, and scale. This whitepaper will focus on Alice & Bob's unique approach to the first of these challenges: developing efficient, fault-tolerant quantum systems to escape decoherence.

// SECOND PART

ESCAPING DECOHERENCE

Quantum Error Correction: The Escape Route

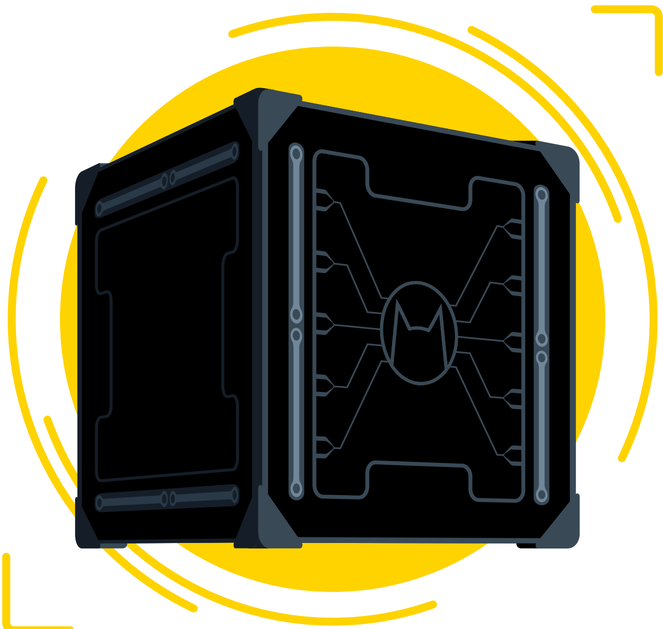
A QUANTUM COMPUTER THAT DECOHERES IS NOT A QUANTUM COMPUTER.

Okay, but what causes decoherence?

Decoherence happens when a qubit has undesirable interactions with its environment. For a person, this may mean that a sudden downpour, a spilled cup of coffee, or a slip on an icy sidewalk damages the clothing they're wearing. For a qubit, this may mean that a tiny influx of thermal energy, a rogue particle, or a faint magnetic field destroys the information it's holding. Although the person gets to stay human, we've got some bad news for the qubit. When a qubit 'decoheres' in this manner, the quantum computer is no longer even quantum. Consequently, the computation produces errors.

To escape this problem, we need to do something that is the opposite of what we're normally told to do. We need to 'think INside the box.'

THIS IS THE FUNDAMENTAL PARADOX OF BUILDING A QUANTUM COMPUTER: HOW CAN YOU CREATE A MACHINE INSIDE AN IMPENETRABLE BOX AND AT THE SAME TIME CONTROL WHAT HAPPENS IN THERE?



Unfortunately, it's impossible to eliminate decoherence. Besides the environment surrounding the qubits, all the communication channels we use to interact with them can cause it.

For this fundamental reason, quantum computers are bound to have much higher error rates than classical computers. This is why much of the story we tell here will be about dealing with decoherence. That is, we accept that there will be errors and we're focused on correcting them. We call this 'quantum error correction.'

THIS IS A VERY IMPORTANT PARADIGM SHIFT, BECAUSE THE GOAL IS NO LONGER TO SACRIFICE OUR CONTROL OVER QUBITS TO PRODUCE AS FEW ERRORS AS POSSIBLE. IT IS FIRST AND FOREMOST TO BE ABLE TO CORRECT THE ERRORS THAT WILL INEVITABLY BE PRODUCED.

Implementing effective quantum error correction is hard and was once considered a pipe dream, however that dream has become reality. A recent steady stream of scientific achievements has made QEC look increasingly plausible[13, 14, 15]. The latest result, from Google[16], definitively demonstrated practical quantum error correction with logical qubits.

The journey is far from over, however. Recent achievements may be remembered as the first times we were able to reduce errors, but we've only seen slight improvements in small quantum systems. We still need to demonstrate significant improvements in large quantum systems.

DECOHERENCE IS STILL THE MAIN BARRIER TO BUILDING USEFUL QUANTUM COMPUTERS, BUT THIS BARRIER IS ERODING.

IN A NUTSHELL

Completely preventing decoherence is impossible. Consequently, we need to be able to fix the errors that inevitably emerge. Once considered a pipe dream, recent advances have made effective quantum error correction increasingly realistic.

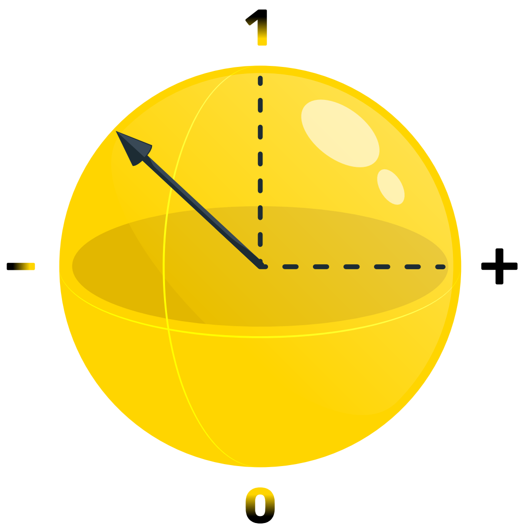
Qubit Basics: From Quantum States to Error Correction

To understand errors in quantum computers, we must first understand the qubit.

First of all, the term 'qubit' has two definitions. It refers both to a single physical device and to the unit of quantum information that is encoded on the physical device.

TO DRAW A PARALLEL WITH CLASSICAL COMPUTERS, THE TERM 'QUBIT' IS ANALOGOUS TO BOTH THE 'BIT' AND THE ELECTRONIC DEVICE HANDLING IT, THE TRANSISTOR.

While we generally refer to a qubit as a physical object, understanding the stickiness of errors requires focusing on the qubit as a mathematical object and the way it encodes information. The most intuitive way to represent the concept of a qubit is by using a tool known as the Bloch sphere, named after the physicist Felix Bloch.



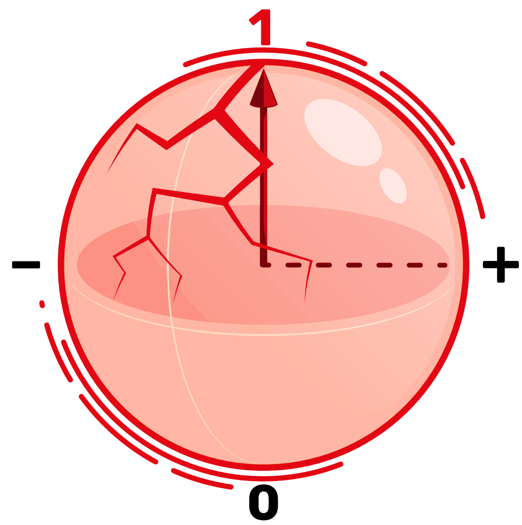
A qubit represented on a Bloch sphere

The state of a qubit can be represented as a point on the surface of the Bloch sphere. The north pole is the classical '1' state, while the south pole is the classical '0' state. All other points on the sphere are in a 'superposition' of '1' and '0.'

When we observe a qubit, a superposition 'collapses' to one of the two poles and we get a classical bit for our troubles: '1' or '0.' The outcome depends on the point's position; if the point was in the northern hemisphere, there is a higher probability we will observe '1,' and vice versa. Along the equator, each outcome is equally likely.

But while the 'North-South' dimension is important, the 'East-West' dimension is really important. This represents the qubit's phase, which is the power core of quantum computing. Specific superpositions of states with opposite phases are represented on the Bloch sphere by the signs '+' and '-.'

The quantum computing industry is all about creating physical devices that behave like this mathematical model. These devices, which are also called qubits, create real superpositions in real physical systems. In a quantum computer, quantum mechanics becomes a reality... but a very fragile one.

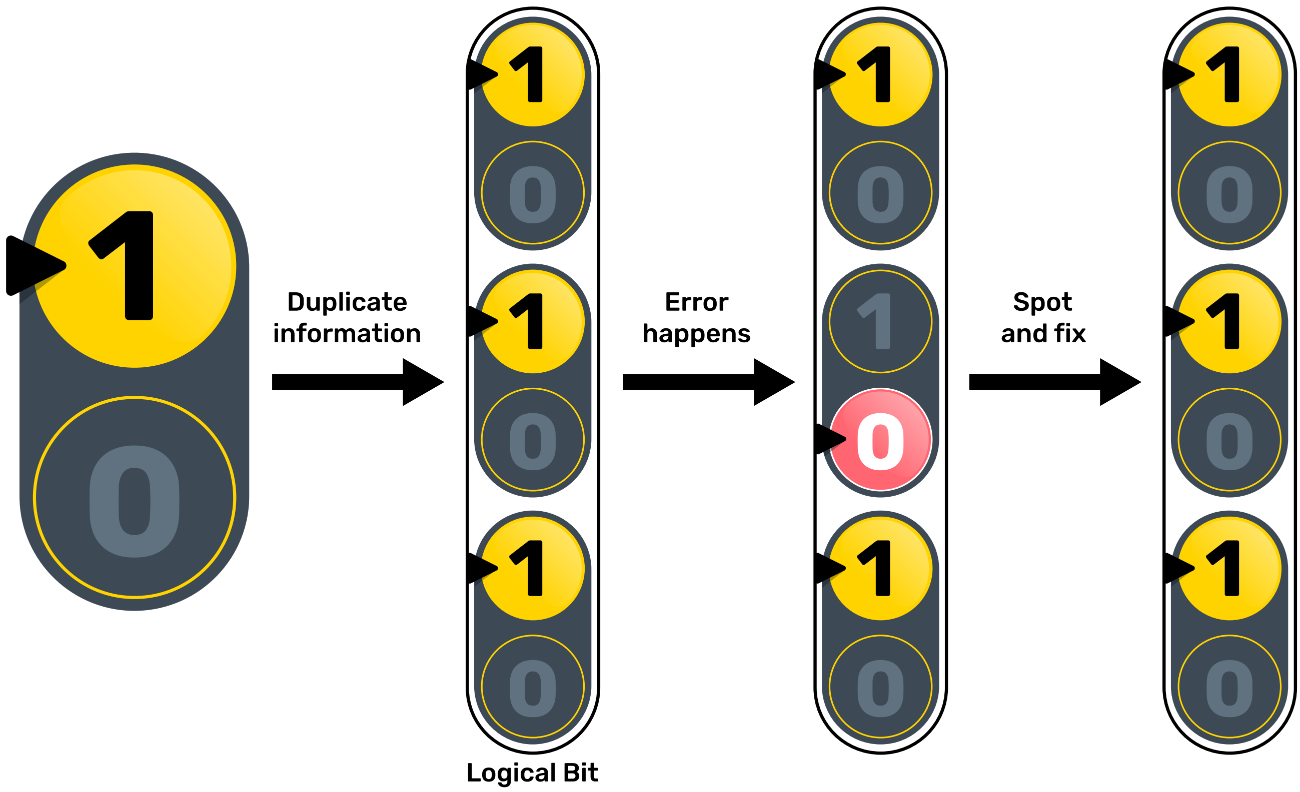


When a qubit decoheres, it loses its quantum information. Its arrow randomly flips to either the north or south pole, which produces errors. A decohered qubit stops following quantum mechanics' delicate properties, such as superposition.

EVERY UNWANTED INTERACTION BETWEEN THE REAL WORLD AND A PHYSICAL QUBIT CAUSES DECOHERENCE AND MAKES ITS STATE COLLAPSE TO ONE OF THE POLES, DESTROYING ALL OUR PREVIOUS EFFORTS.

Classical computers are not immune to errors. Errors are common, in fact, in long-distance and wireless communication. These errors can be corrected, however, by using error correction codes. We've been working on perfecting such codes since the 60s, and they exploit a simple principle: redundancy. We'll illustrate, as an example, the simplest of such codes: the repetition code.

CLASSICAL ERROR CORRECTION



The repetition code is used in classical computing to protect information in very noisy channels. This example shows a simple case where each bit is redundantly encoded three times. If one of the bits is corrupted and flips, it can be identified through a majority vote. The corrupted bit can then be flipped back, correcting the error.

The principle is similar to voting. A one might be represented by 111, as shown above. If 101 is detected, the ones outvote the zero, and it is corrected back to 111. The repetition of the ones as the correct value protects against errors in individual bits. Collectively, these bits form a 'logical bit.'

Spotting a corrupted bit in a logical bit is supposedly easy: which bit is unlike the others? Unfortunately, there is always a chance that multiple bits were corrupted: 111 becomes 001. Luckily, we can exponentially decrease the probability of an error simply by using more bits: 11111 can become 11001 and still be fixed. Adding just a few bits can dramatically improve reliability.

ERROR CORRECTION, THEREFORE, TRADES QUANTITY FOR QUALITY. WE TAKE MULTIPLE ERROR-SUSCEPTIBLE BITS AND FORM A HIGH-QUALITY LOGICAL BIT.

Fortunately, we can do the same thing with qubits. We can take multiple error-prone physical qubits and form high-quality 'logical qubits.' Qubits are a bit more complex than bits, however; they have three features that turn QEC into a major engineering challenge.

IN A NUTSHELL

We need qubits, the building blocks of quantum computers, to handle quantum information. Unfortunately, qubits are fragile and easily disrupted by environmental interactions, which cause errors. Quantum error correction should be able to fix these errors, but three qubit-specific features make this a significant engineering challenge.

Qubit Feature 1: At Least Two Types of Quantum Errors

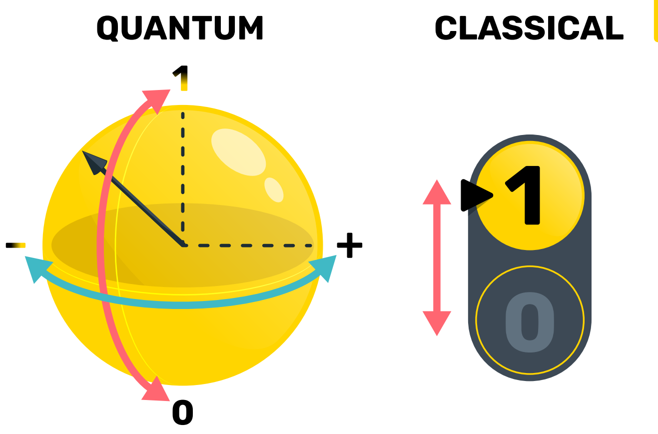
When a qubit decoheres, its state collapses to a random binary value, 1 or 0, and the information it carries is lost. Our quantum computer turns into a very expensive random number generator.

Two types of errors are produced this way:

- **Bit-flip errors** exchange the '1' and '0' states (North and South on the Bloch sphere).
- **Phase-flip errors** exchange the '+' and '–' states (East and West on the Bloch sphere).

Classical bits suffer from bit-flips, but they don't have phases, so they don't experience phase-flips.

Also, two types of errors are actually the best-case scenario. Most superconducting qubits suffer from leakage errors, which cause qubits to mistakenly jump to '2' or higher states, instead of staying in superpositions of '1' and '0.' Photonic, neutral atom, and trapped ion qubits, meanwhile, are capable of physically escaping the computation. Instead of being in error, per se, they're physically gone.



In quantum computing, not only are errors much more frequent due to the fragile nature of quantum information, but we need to deal with two types of errors instead of one.

IN A NUTSHELL

The first qubit-specific feature we have to contend with is that qubits experience at least two types of errors, whereas bits only experience one. Both experience bit-flips (switching '1' and '0'), but only qubits experience phase-flips (switching '+' and '-'). Unlike classical error correction, quantum error correction is a multi-dimensional challenge.

Qubit Feature 2: Measuring Means Losing

Correcting errors on classical computers can be as simple as reading redundant bits and performing a majority vote. This process determines which bits have been flipped so we can flip them back and return our logical bit to a valid state.

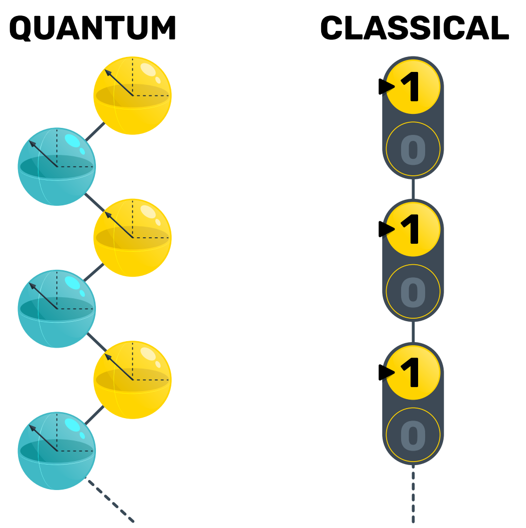
We would like to do this with our qubits, but they demand privacy. We can't directly observe them without corrupting the information they're holding. From pop culture, you might remember the story of Schrödinger's famous cat:

> **THE OBSERVER CHANGES QUANTUM REALITY THROUGH THE SIMPLE ACT OF 'OPENING THE BOX' AND LOOKING INSIDE IT.**

We can give our sensitive qubits some of the privacy they seek by working with a second set of qubits:

• **Our original set of 'data qubits'** holds the redundant quantum information and must not be disturbed.

• **A new set of 'auxiliary qubits'** acts as 'sentinels,' surveying our data qubits without disturbing them, thus not destroying the information they're holding.



The need for auxiliary qubits typically doubles the hardware resources required for a quantum computer to implement the same error correction code as a classical computer.

Auxiliary qubits are a necessary evil. We need them to make quantum error correction work, but they further increase the number of qubits we need to perform reliable computation.

IN A NUTSHELL

The second feature is that qubits cannot be directly measured without losing their information. Quantum error correction must use auxiliary qubits to monitor our data qubits, because they can do so without disrupting them. This works, but the cost is additional complexity and a further increase in the number of qubits we need.

Qubit Feature 3: The Quality Threshold of Error Correction

We mentioned that quantum error correction faces three major challenges, but 'three' deserves an asterisk because one of these challenges is the error-correction process itself. It introduces a new class of errors as an unintended consequence.

Classical computing has it easy, because storage is resilient and reading operations are extremely reliable. If we measure that a bit has flipped, the most likely explanation is that the bit has indeed flipped.

IN QUANTUM COMPUTING, HOWEVER, ERROR CORRECTION IS A DOUBLE-EDGED SWORD.

The operations involved in detecting errors have low fidelity and they run on fragile qubits. So, when we detect that a qubit has flipped, maybe it flipped, but maybe our error detection operations malfunctioned.

Because of this, our endgame can't rely exclusively on having lots of physical qubits. These physical qubits also have to be 'good' enough.

IT CAN BE PROVEN, IN FACT, THAT THERE IS A THRESHOLD FOR THIS. THERE IS A LEVEL OF QUALITY THAT PHYSICAL QUBITS MUST REACH BEFORE QUANTUM ERROR CORRECTION CAN REMOVE ERRORS.[17,18,19,20]

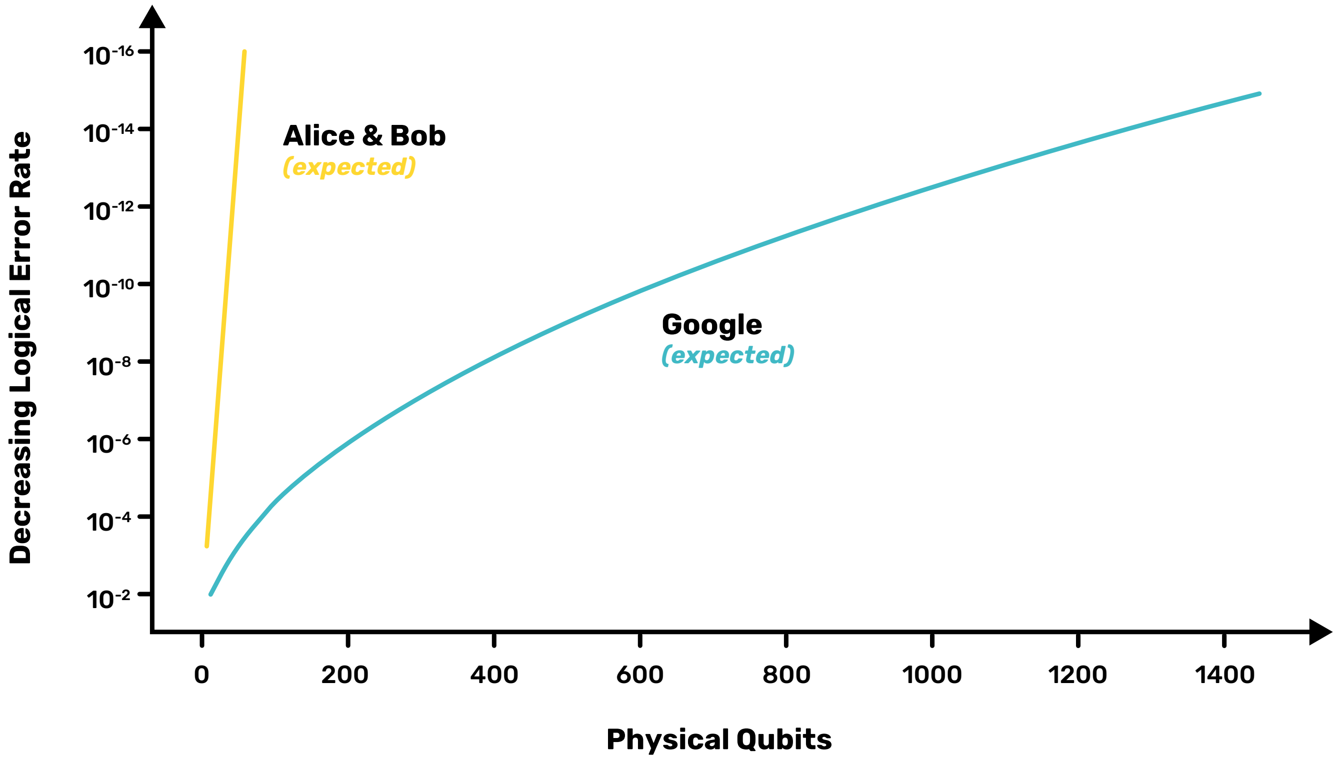
1. **Above threshold,** adding physical qubits actually makes the problem worse!

2. **At threshold,** adding physical qubits does nothing. Reaching threshold will be a significant scientific achievement, but it will only be an intermediate step on the road to practical error correction.

3. **Below threshold,** adding physical qubits finally improves the fidelity of our logical qubit. As a reward for making it this far, the improvement is significant. Each additional qubit exponentially improves the logical error rate, making it finally possible to reach the very low error rates we need. If our error rate is still not low enough to escape decoherence, we can simply add a few more qubits!

Make no mistake, reaching threshold will be a challenge and is a major goal for most of the quantum computing industry. At the time we're writing this, only Google has demonstrated below-threshold physical qubits, which they then used to implement error correction.

ERROR RATE vs QUBIT COUNT FOR LOGICAL QUBITS



The plot shows how error rates for the logical qubits of Google and Alice & Bob architectures decrease exponentially as more physical qubits are included in the error correction code. With similar assumptions, with the surface error correction code, Google, in blue, would require more than a thousand physical qubits to reach a 10^{-15} logical error rate. Alice & Bob's projections, in yellow, indicate that 53 physical cat qubits would be enough to reach the same fidelities using our linear repetition code.

We at Alice & Bob target 99.9999% fidelity for an early logical qubit. This is the measure of the reliability and accuracy of a qubit. In plain language:

THIS MEANS 1 ERROR FOR EVERY 1 MILLION OPERATIONS. WE CALL THIS A '6-NINES LOGICAL QUBIT.'

For quantum computers to deliver on their full promise, the reliability of logical qubits will have to grow to at least 10-nines, and arguably more. Until then, 6-nines will be a sturdy platform for scaling up our systems.

IN A NUTSHELL

The third qubit-specific feature is that they are error-prone by nature. For quantum error correction to work, our qubits must exceed a quality threshold. Once we are below this threshold, adding qubits exponentially improves fidelity. In the near term, we at Alice & Bob aim to achieve a '6-nines' logical qubit, which means 1 error for every 1 million operations.

Error Correction's Immense Cost: The Surface Code Example

One of the most widely used quantum error correction codes is called the 'surface code'[21]. Its two-dimensional grid structure has squares of data qubits, with auxiliary qubits in the middle of each square checking whether the data qubits agree or not. Half of the auxiliary qubits check for phase-flip errors, while the other half check for bit-flip errors.

The surface code has many qualities that have helped make it an industry-standard, but it also has a major shortcoming. When we want to increase its ability to detect and correct errors happening simultaneously on multiple data qubits, the number of qubits we need scales quadratically.

'Quadratic' scaling means:

- Protecting against any **2 simultaneous errors** requires **49 qubits.**
- Protecting against any **3 simultaneous errors** requires **97 qubits.**
- Protecting against any **4 simultaneous errors** requires **161 qubits.**

...

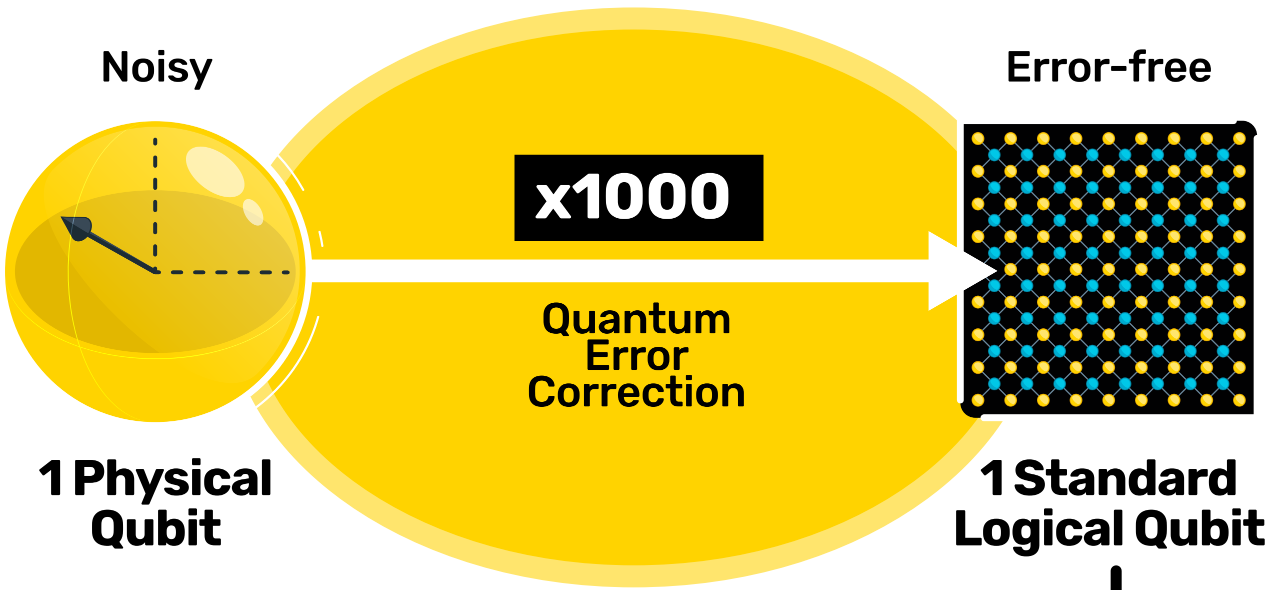
For enough protection to start running some algorithms, we'll need around 449 to 1,249 physical qubits per logical qubit. But to run large algorithms like Shor's algorithm, studies have shown that we'll need around 1,000 physical qubits per logical qubit[22].

THESE NUMBERS ARE FOR ONLY ONE LOGICAL QUBIT. MOST CURRENT QUANTUM PROCESSORS DON'T EVEN HAVE ENOUGH PHYSICAL QUBITS TO CREATE EVEN ONE OF THESE!

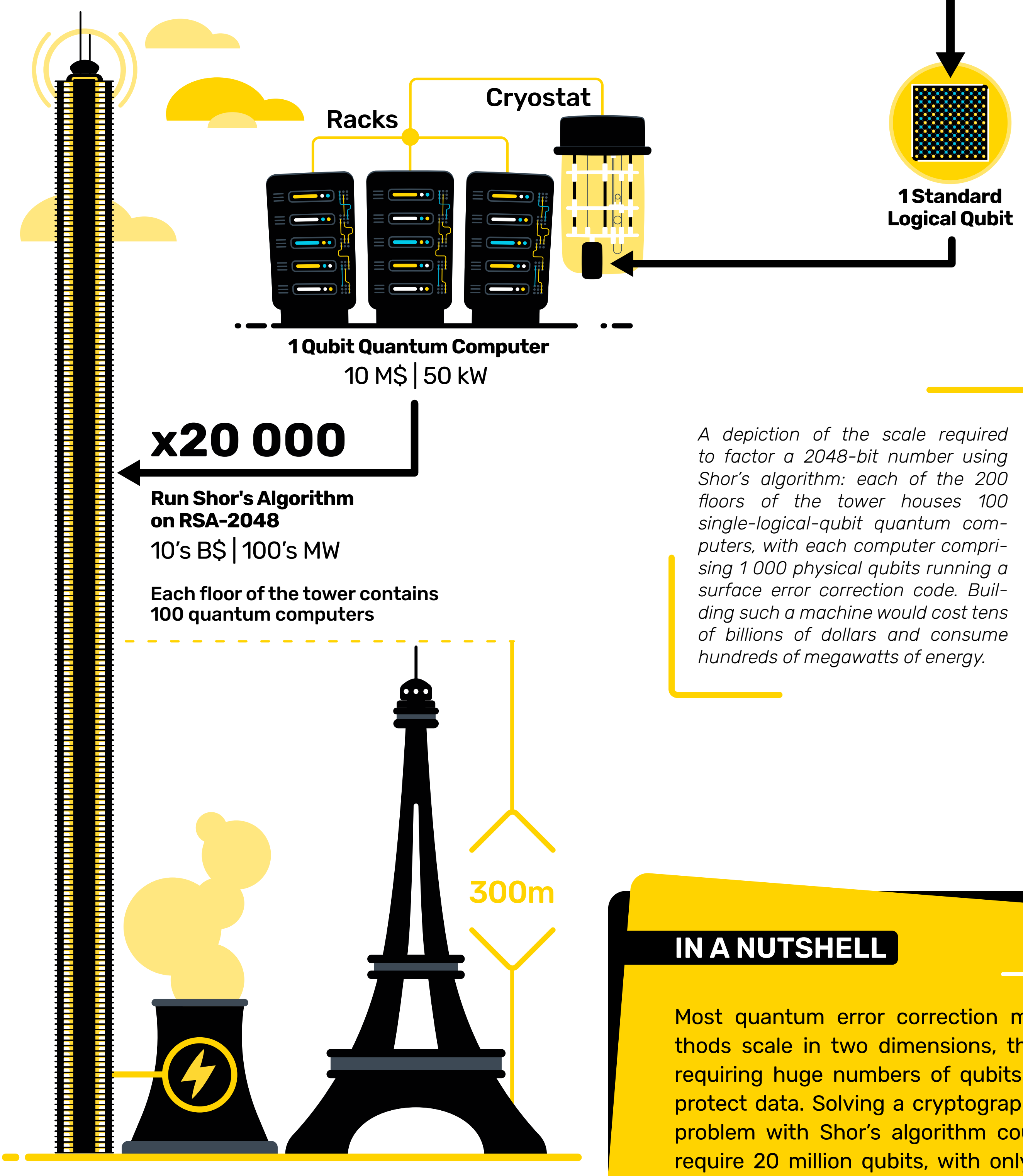
We also have to take a step back and look at the bigger picture. Not only we need to create logical qubits, but we also need to create the immense infrastructure to perform logical gates on these logical qubits.

Let's use, as an example, common transmon qubits with a surface code to factor a number with 2,048 binary digits. This is notoriously a classically intractable cryptography problem, but we can solve it with Shor's algorithm and 20 million physical qubits. To give you an idea of the 'overhead' of error correction and logical gates, only 0.06% of the qubits in this setup are used to perform computation; the others are there only to correct errors.

THE HARDWARE NEEDED TO RUN SHOR'S ALGORITHM



Using the surface error correction code, we can create logical qubits, but this requires a substantial number of additional physical qubits and supporting hardware. Depending on the fidelities we aim to achieve, thousands of physical qubits might be necessary.



A depiction of the scale required to factor a 2048-bit number using Shor's algorithm: each of the 200 floors of the tower houses 100 single-logical-qubit quantum computers, with each computer comprising 1 000 physical qubits running a surface error correction code. Building such a machine would cost tens of billions of dollars and consume hundreds of megawatts of energy.

Since most quantum computers, again, don't have enough physical qubits for even one high-fidelity logical qubit, it's safe to characterize this scenario as way beyond today's capabilities. And the challenge extends beyond the qubits themselves to cabling, cryogenics, and all the other qubit-supporting components. ■

IN A NUTSHELL

Most quantum error correction methods scale in two dimensions, thus requiring huge numbers of qubits to protect data. Solving a cryptographic problem with Shor's algorithm could require 20 million qubits, with only a fraction of a fraction of those qubits performing the computation. Today's largest superconducting quantum computer can pack just over a thousand qubits onto a chip, so we're not quite there yet.

An Alternative Approach to Error Correction: Hardware Efficiency

A working surface code has been demonstrated on a real quantum computer, but scaling it is going to be difficult. Even the authors of Google's paper confirm this[16].

To build our target prototype of a single 6-nines logical qubit, we would need 1,457 physical qubits using Google's architecture. This is the effect of the surface code's quadratic scaling, given the hardware's high physical error rates[16].

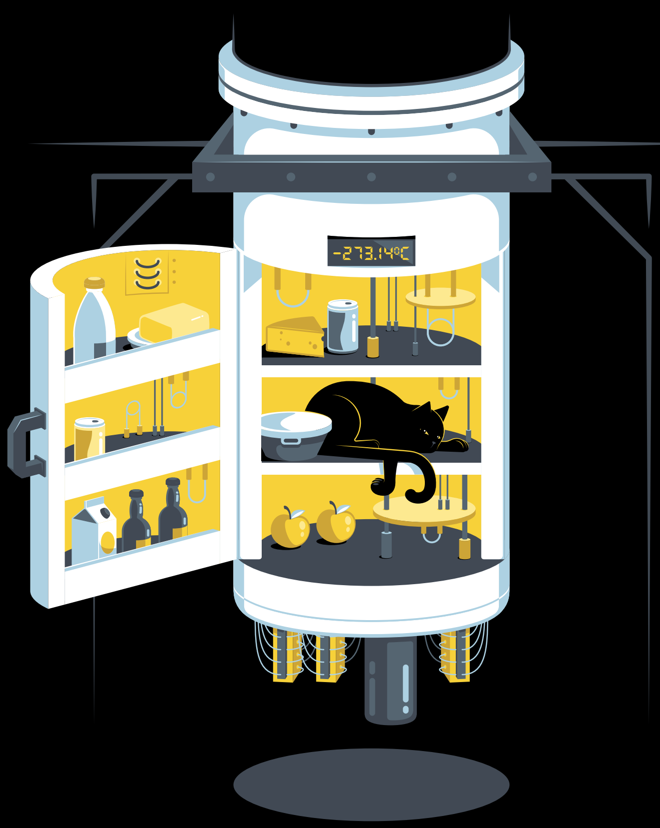
The largest superconducting quantum processor today, however, is IBM's Condor chip with only 1,121 physical qubits. While this is three-quarters of the way there numerically, Condor does not have an error correction architecture.

At Alice & Bob, we've always been focused on the importance of quantum error correction and the scaling challenges that come with it. This is why, from the very beginning, we've invested all our efforts in one of the most hardware-efficient technologies for quantum error correction: the cat qubit.

Okay, but what do we mean by 'hardware-efficient'?

> **SIMPLY PUT, A HARDWARE-EFFICIENT APPROACH NEEDS FEWER PHYSICAL QUBITS TO ENCODE A LOGICAL QUBIT COMPARED TO OTHER APPROACHES.**

Over the long-term, a hardware-efficient approach may be the only viable way to build a large-scale, error-corrected system. This is the key to driving down hardware costs, reducing operational complexity, minimizing energy consumption, and keeping the physical sizes of these systems plausible.



IN A NUTSHELL

A hardware-efficient approach may be the only practical way to build large-scale, error-corrected quantum computers. The cat qubit exemplifies such an approach, requiring fewer physical qubits per logical qubit than other approaches. Cat qubits reduce the hardware costs, operational complexity, energy consumption, and physical footprint of a scaled system.

// THIRD PART

CAT QUBITS: THE SHORTCUT TO PRACTICAL ERROR CORRECTION

From 2D to 1D: How Cat Qubits Change Quantum Error Correction

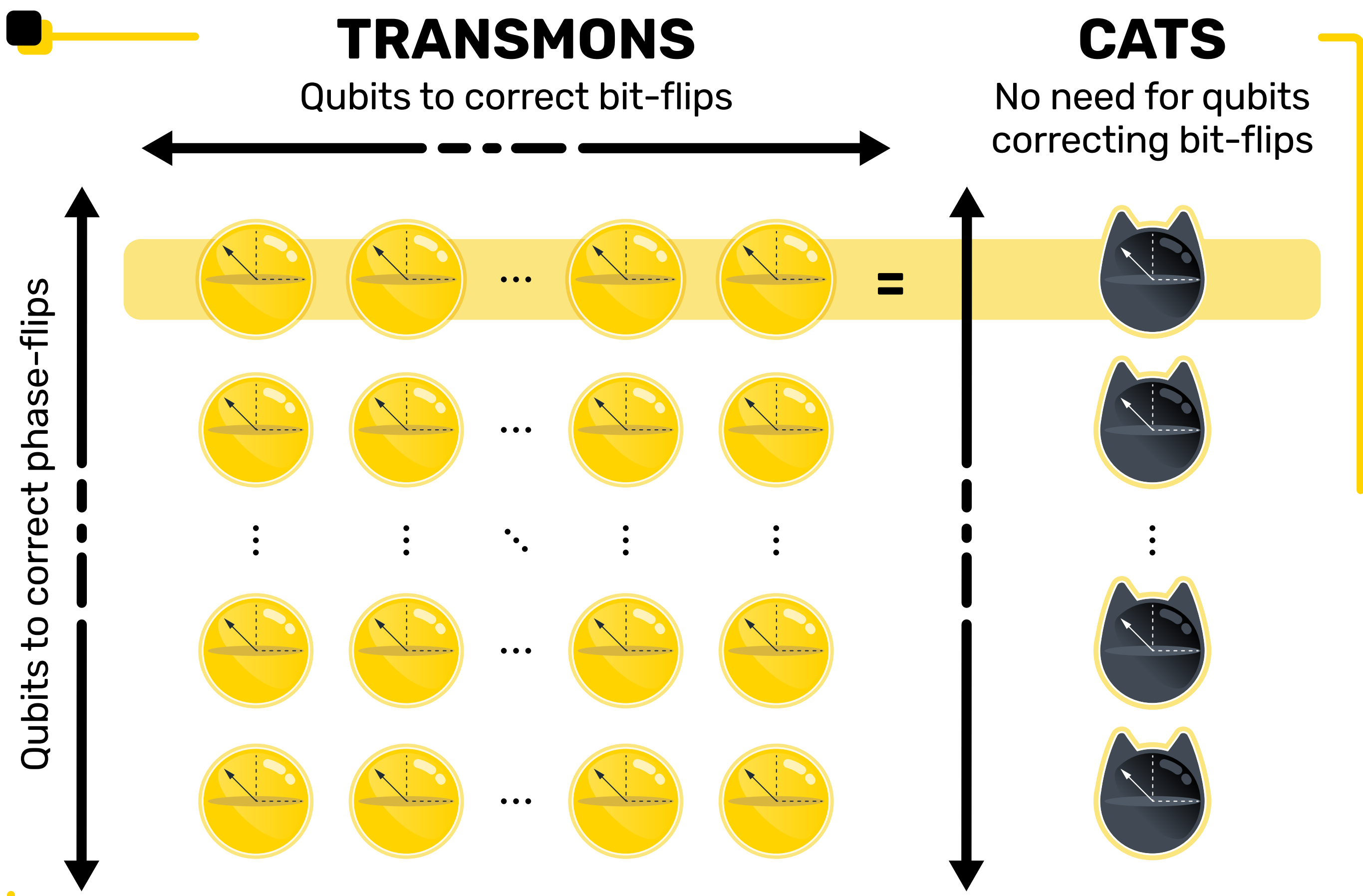
What if we could level the playing field? Handling two types of errors simultaneously is so hard that today's largest quantum computers, using a surface code, could barely encode one high-quality logical qubit. It's as if classical computers placed their goal on top of a mountain while quantum computers placed their goal on a flat, grassy field. Team Quantum is at a laughable disadvantage.

But, again, what if we could level the playing field? What if we could reduce quantum error correction to 1D, like classical error correction, instead of 2D? What if we could move the other team's goal from the mountaintop to the same grassy field?

Fortunately, this is possible. And this is exactly what cat qubits do. They can be tuned to exponentially reduce bit-flips, up to a point where bit-flips go virtually extinct.

AT THIS POINT, WE ONLY NEED TO DETECT AND CORRECT PHASE-FLIP ERRORS, MAKING QUANTUM ERROR CORRECTION A 1D PROBLEM.

The cat qubit is a relatively young quantum bit design. Cat states have been a promising theoretical concept for years, but engineering the control hardware for them proved challenging. The error-correction properties of cat qubits were demonstrated for the first time in 2020, thanks to the pioneering work by Alice & Bob's founders[23].



The surface code shown on the left encodes quantum information redundantly in two dimensions: one to correct bit-flips and the other to correct phase-flips. A cat qubit architecture, in contrast, would require only one dimension, achieving error correction with a fraction of the resources, as shown on the right.

The successful experiment led to the formation of Alice & Bob, which has been improving the cat qubit over the past five years. Fast forward to today, cat qubits have become a main contender in the race to build a universal fault-tolerant quantum computer.

You may be wondering why they're called 'cat' qubits, and this is a reference to Schrodinger's famous thought experiment. Cat states are superpositions of very different quantum states which are not strictly macroscopic but display almost-classical behavior. This, of course, resembles Schrödinger's cat, a classical system in a superposition of two very different macroscopic states!

To create a cat qubit and protect its information from bit-flip errors, we use a so-called coherent state of light (photons) in a superconducting chip.

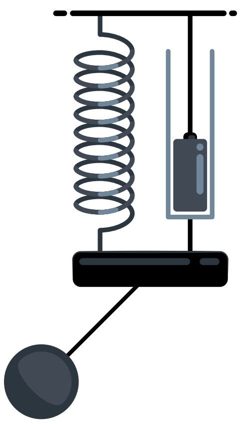
IN A NUTSHELL

Cat qubits simplify quantum error correction by reducing the problem from two dimensions to one. Bit-flips are driven to the brink of extinction, leaving only phase-flips to contend with. Compared to surface codes, this approach offers a much more efficient path to fault-tolerance.

More Than a Pendulum: Cat's Autonomous Stabilization

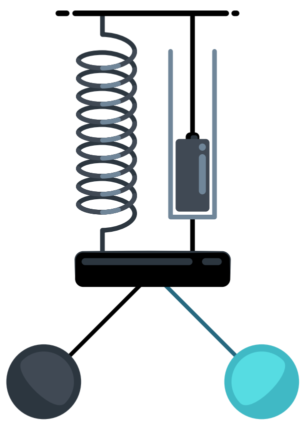
What is a 'coherent state'? A coherent state is an oscillating electric field that behaves like a classical harmonic oscillator. Imagine a swinging pendulum, and you're almost there.

After we activate a pendulum, its swings gradually diminish in height. If we do nothing more, it will eventually stop. We essentially want a pendulum that remains stable over time, so we add a driving force to a spring, which maintains the swinging motion of our oscillator.



Controlling this force isn't easy. The problem is that if we add too much energy, the oscillations will get wider and wider, eventually swinging out of control. On the other hand, if we add too little energy, the oscillations will eventually die out. The solution is to add another component, a damper, which dissipates any excess energy in the spring.

WITH A SPRING AND A DAMPER BOOSTING AND MITIGATING THE OSCILLATIONS, RESPECTIVELY, THE PENDULUM GRADUALLY REACHES A STABLE MOTION. THIS IS THE KEY TO ENCODING OUR INFORMATION.



A qubit needs to be in a superposition of two distinct, stable states, 1 and 0. Using this pendulum analogy, these two stable states correspond to two identical pendulums swinging at the same frequency and amplitude but in exactly opposite directions. Imagine two pendulums starting at the same moment, but one starts to the left while the other starts to the right.

A remarkable phenomenon that emerges out of this system is that it quickly discards all other states. If we disturb our 'pendulum,' its stabilization mechanism soon brings its movement back to one of the two stable oscillations.

If we want to change the state intentionally, we have to 'kick' it relatively hard. A bit-flip, continuing this analogy, kicks a pendulum so hard that its oscillations reverse. If it was the pendulum that originally swung to the right, it synchronizes with the pendulum that originally swung to the left, and vice versa.

THIS, THEREFORE, IS THE KEY INTUITION BEHIND CAT QUBITS. THE JOINT ACTION OF THE SPRING AND THE DAMPER PROTECTS EACH PENDULUM'S STATE AGAINST MOST DISTURBANCES.

Only rare, big events can force a pendulum to errantly switch to the opposite state and cause a bit-flip.

Interestingly, we can't tell how a pendulum is swinging based solely on observing the spring and the damper. We can't peek accidentally into the box, thus preserving our quantum superpositions.

While pendulums are useful analogies, we cannot use them to build a quantum computer. Fortunately, though, we can implement a similar principle. We can inject photons into a superconducting resonator, a tiny tab of aluminium and tantalum on a printed circuit.

IN A NUTSHELL

The cat qubit uses a stabilization mechanism analogous to a pendulum, where a 'spring and damper' system ensures stable oscillations. This autonomous system protects the pendulum's oscillations from inverting, thereby preventing bit-flip errors.

Bit-Flips Protection Record: The Noise-Biased Cat Qubit

Injecting photons into a cat qubit is the spring in our pendulum analogy. Adding photons increases the 'distance' between the two states, as if the pendulum swings get higher. This 'distance' suppresses unwanted jumps (bit-flip errors) between the two possible oscillations, making the states resilient to external noise. And if that's not interesting enough for you, try this on for size: the protection against bit-flip errors grows exponentially with the number of photons. This protection becomes significant with just a few photons.

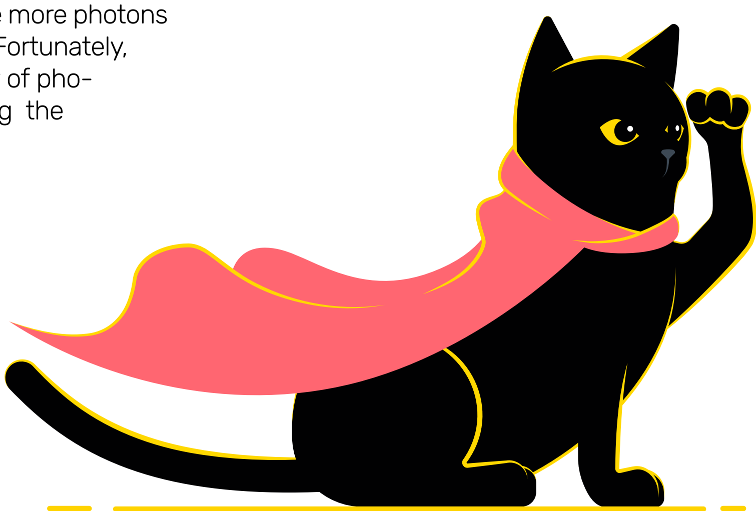
What about phase-flips? These are jumps between the even and odd cat states, '+' and '-,' which we can't visualize with our pendulum analogy. The difference, intuitively, is the number of photons. If we lose a photon, an even number of photons becomes odd, and vice versa. Increasing our photon count to suppress bit-flip errors introduces a higher risk of phase-flip errors, because the more photons we have the more likely we are to lose one. Fortunately, the increase is only linear with the number of photons, an acceptable trade-off considering the exponential decrease in bit-flip errors.

WE CAN SAY, THEREFORE, THAT A CAT QUBIT BENEFITS FROM A STRONG 'NOISE BIAS.' THIS MEANS THAT IT TAKES MUCH LONGER FOR A BIT-FLIP ERROR TO OCCUR THAN A PHASE-FLIP ERROR. THIS BIAS IS TUNABLE BY CHOOSING THE AVERAGE PHOTON NUMBER.

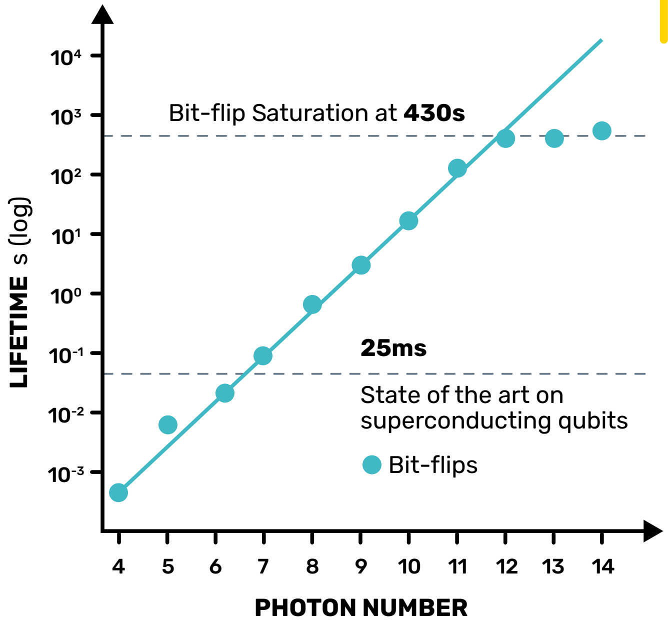
What are our experimental results? We're glad you asked. Our latest single cat qubit chip, called Boson 4, can average up to seven minutes between bit-flip errors with just 12 or more photons.

For reference, the best-case scenarios for a typical superconducting qubit are measured in milliseconds.

IF WE DO A SIDE-BY-SIDE COMPARISON, WE SEE THAT OUR CAT QUBITS PROTECT INFORMATION FROM BIT-FLIP ERRORS MANY TENS OF THOUSANDS OF TIMES LONGER THAN OTHER SUPERCONDUCTING QUBITS. WE'RE ALMOST ELIMINATING THEM ENTIRELY.



THE NOISE BIAS



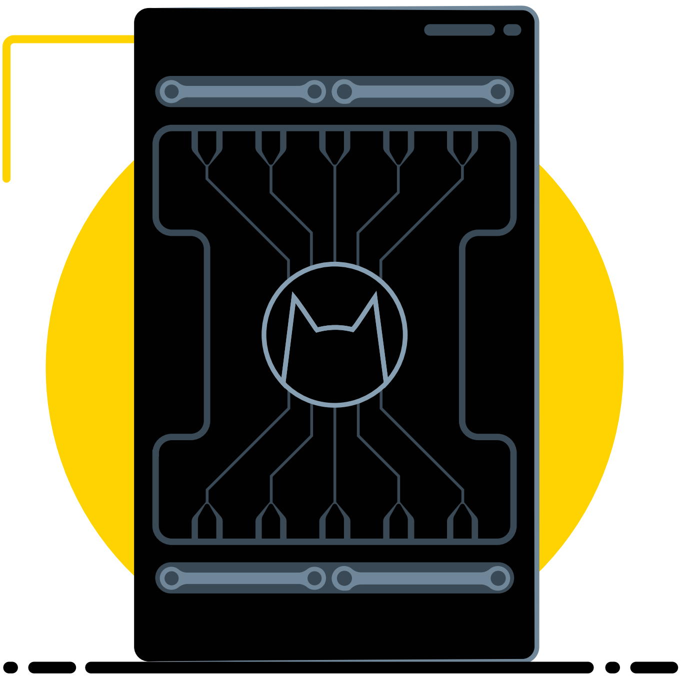
Exponential suppression of bit-flips on the Boson 4 chip. The bit-flip time (vertical axis, log scale) is measured (blue dots) as a function of the mean number of photons in the memory (horizontal axis). The bit-flip time increases exponentially (tilted blue line) before saturating around 430 seconds.

To reap benefits from this, we must design all subsequent operations on the cat qubit to preserve this noise bias and not reintroduce bit-flips. Fortunately, our theorists have shown that this can be done[24].

IN A NUTSHELL

With each additional photon, cat qubits offer exponentially stronger protection against bit-flip errors. However, the trade-off of photon injection is a linear increase in phase-flip errors. This favorable trade-off is known as noise bias. This year, our Boson 4 chip achieved a bit-flip lifetime of up to seven minutes, a world record for superconducting qubits.

The Hardware Behind Cat Qubits



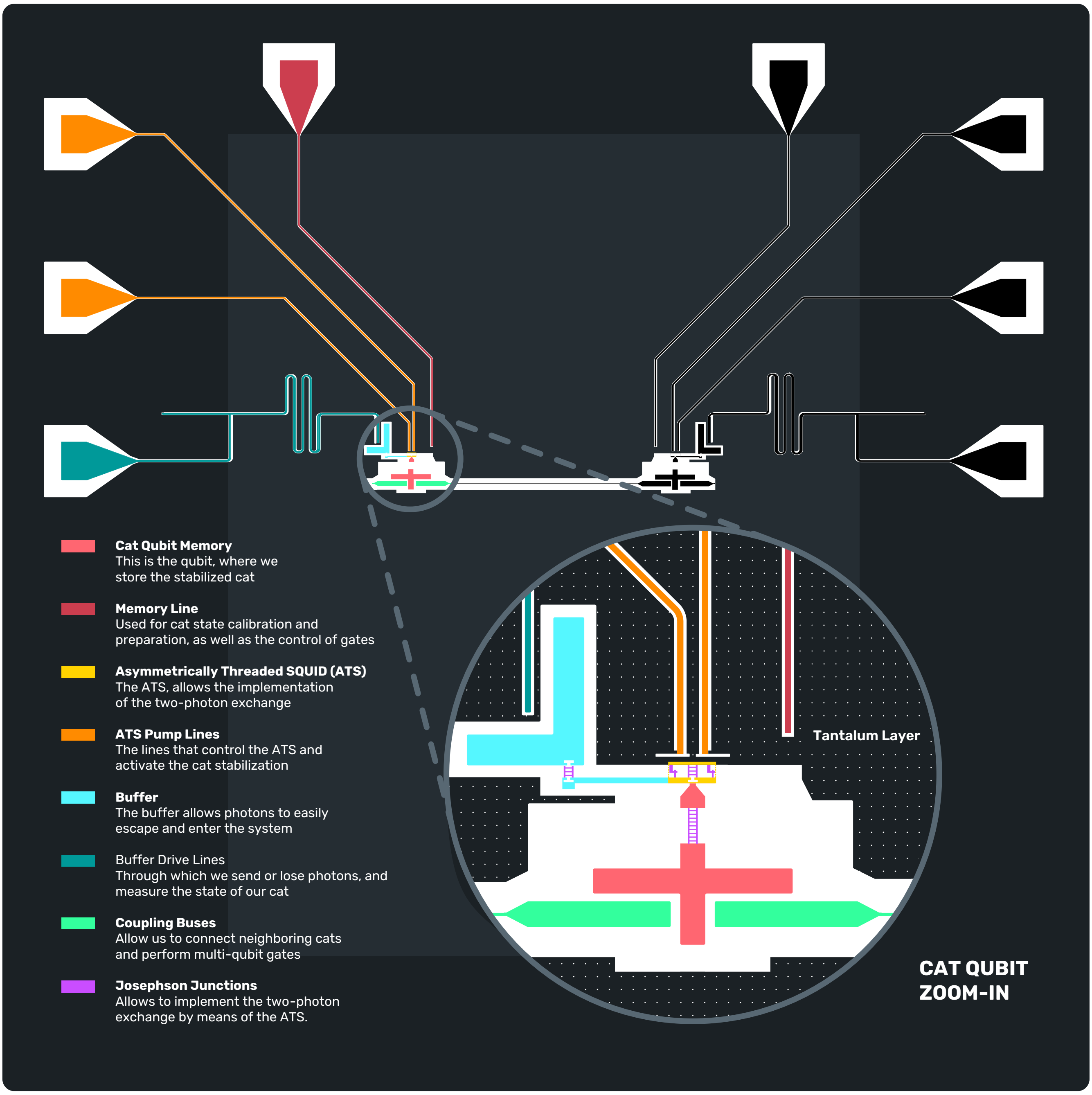
At Alice & Bob, we create and improve the quantum computer's brain, the quantum processor. We are first and foremost a quantum processing unit (QPU) designer and manufacturer. Although it might sound like we have swinging pendulums on our chips, we assure you that we only use superconducting resonators and circuits.

That said, how do we create qubits that work like stabilized pendulums? What is the quantum version of this idea?

Well, let's start with the chip.

A CAT QUBIT IS A SAPPHIRE CHIP WITH A SMALL APPARATUS OF PHYSICAL SYSTEMS MADE MOSTLY OF SUPERCONDUCTING METALS.

A CAT QUBIT ZOOM-IN



A zoom-in of one of our quantum processors. This chip measures 1 cm on each side and features two cat qubits disposed symmetrically on the top half. The legend provides descriptions of all the key elements on the chip and their functions.

The cat qubit encodes information in a quantum harmonic oscillator, which is set up in a superconducting resonator, a piece of tantalum and aluminum several hundred micrometers wide. This is the memory of the cat qubit, where quantum information is stored in the collective wave-particle behavior of the individual photons trapped within the metal.

The main source of error in such a memory is photon loss, environmental noise causing single photons to escape the resonator. Unless something is done about it, every state in our memory will ultimately converge to the vacuum state, which is a resonator with no photons in it. This is analogous to a still pendulum.

This is why our cat qubits are engineered to allow two-photon injection and dissipation, our 'quantum spring' and our 'quantum damper,' respectively. In fact, we created an ingenious, if we do say so ourselves, non-standard situation where our memory gains or loses photons in pairs.

This engineered interaction exchanges pairs of photons in our resonator with one photon from an intentionally lossy element we call the buffer, where it is eventually dissipated into the environment. This generates and stabilizes the cat's two coherent states.

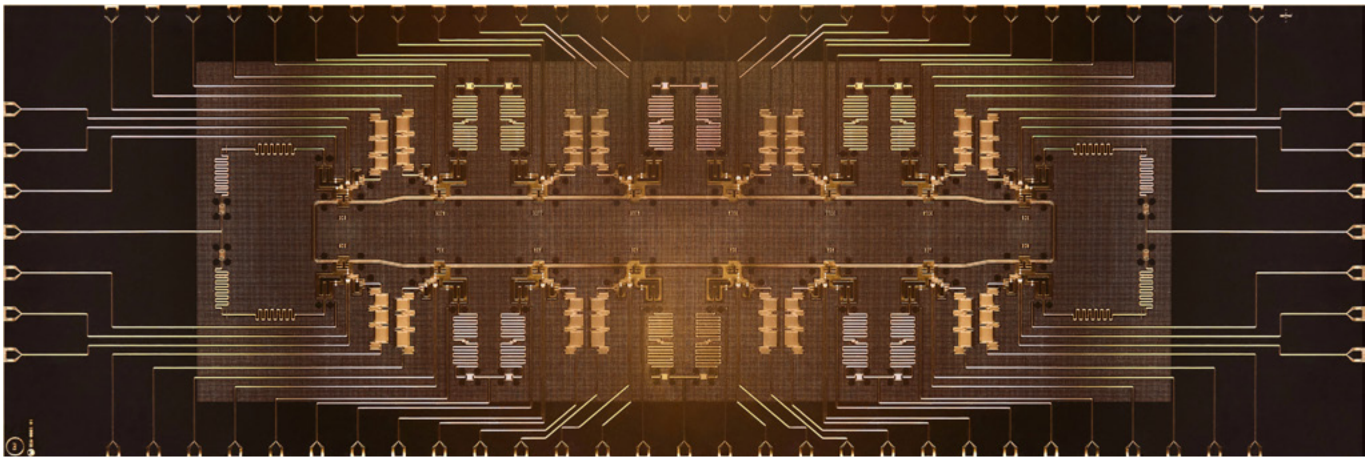
Why do we use pairs of photons? As previously mentioned, changing the number of photons in our memory from even to odd, or vice versa, changes the phase of our qubit: a phase-flip error. By only allowing pairs of photons in and out of our memory, we avoid this change of parity.

Finally, to ensure optimal performance and stability, we place our entire system into a dilution refrigerator and cool it down to 10 millikelvin. This extreme cold temperature is crucial for maintaining a complete energy vacuum and shielding our cat qubits from thermal noise.

IN A NUTSHELL

Our quantum processors consist of sapphire chips with printed superconducting circuits, where quantum information is stored in coherent states of photons. Our system allows for the injection and dissipation of photons, which stabilizes our qubit. Our entire system operates at 10 millikelvin within a dilution refrigerator, shielding it from thermal noise.

HELIUM



A close-up of our Helium chip, a 3 cm quantum processor featuring 16 cat qubits connected in a looped line. Despite its compact size, this processor has 72 ports, each connected to a cable.

Cats' Hardware Efficiency: Repetition Code and LDPC Connectivity

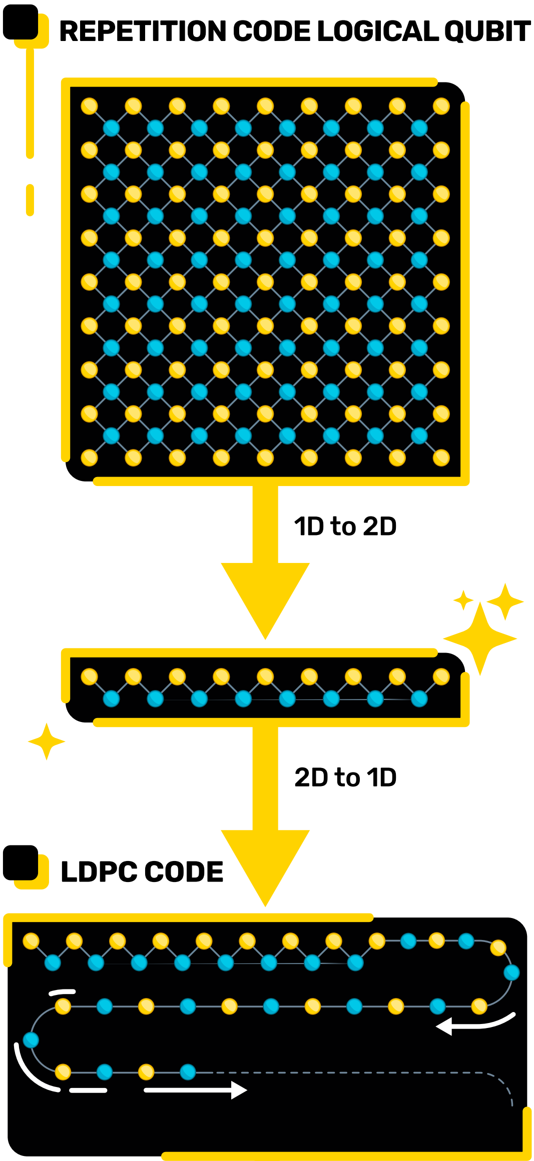
It's better to work with only one error because it allows us to build our error correction in one fewer dimension, 1D vs. 2D. This leaves one more dimension free for other uses. Qubits are, after all, physical systems that need to be arranged in an 'architecture' that allows us to store information, run operations, and correct errors.

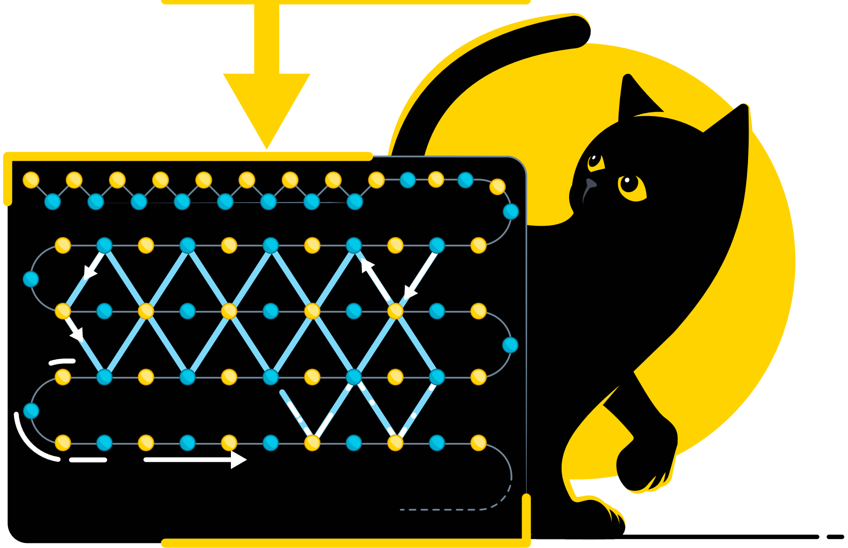
THE MAIN BENEFIT OF DEALING WITH ONE DIMENSION OF NOISE, INSTEAD OF TWO, IS THAT IT ALLOWS US TO USE ERROR CORRECTION ARCHITECTURES THAT REQUIRE FEWER QUBITS.

As we mentioned earlier in this whitepaper, 2D error correction codes, such as the surface code, arrange our physical qubits, data and auxiliary, into a grid. However, cat qubits, which only have to handle phase-flip errors, can be arranged into a 1D line. By alternating data and auxiliary qubits, we can use the classical error correction code we encountered earlier, the repetition code, to correct phase-flip errors.

A repetition code is far more economical than a surface code. Remember the 20 million physical qubits we need to run Shor's algorithm?

WE CAN REDUCE THE NUMBER OF QUBITS WE NEED TO RUN SHOR'S ALGORITHM BY 60 TIMES. THAT'S APPROXIMATELY 360 THOUSAND[25] CAT QUBITS USING A REPETITION CODE, COMPARED TO 20 MILLION STANDARD SUPERCONDUCTING QUBITS USING A SURFACE CODE.





A 98.3% reduction is hopefully impressive, but we might be able to do even better with a low-density parity-check (LDPC) code. LDPC creates extra connections between distant qubits, allowing us to get more out of them. If we were using a 2D error correcting code, applying LDPC would require a third dimension for these extra connections. A 3D architecture would be even more challenging to engineer than a 2D architecture and would also make it more challenging to implement gates[26].

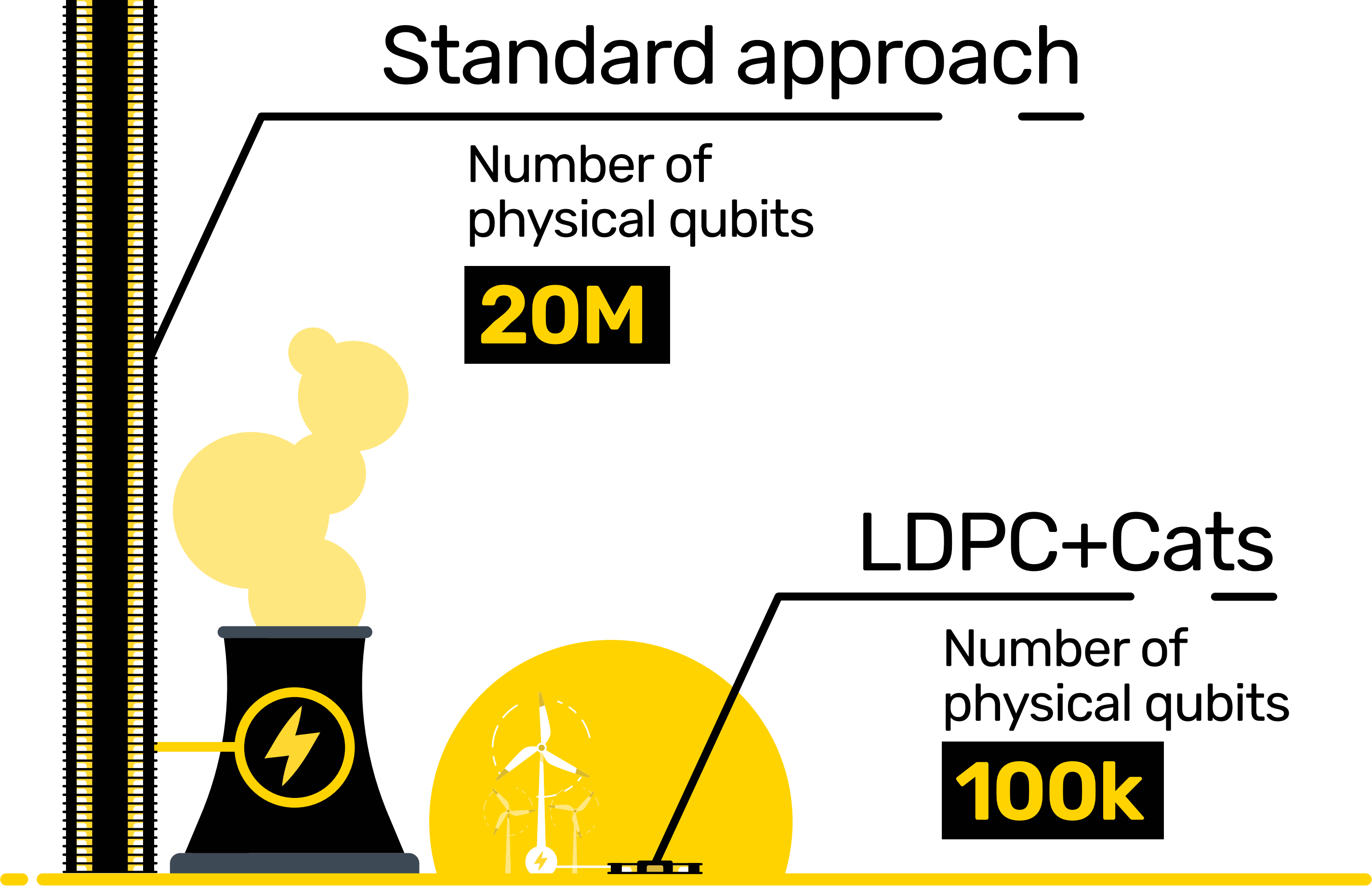
Cat qubits are arranged in a line that can be folded in such a way that distant qubits become neighbors, making them easier to connect. With this architecture, we have found a pattern of cat qubits that allows us to implement LDPC in only two dimensions. A 2D LDPC code requires up to five times fewer qubits per logical qubit than our previous best code, while still allowing the efficient implementation of gates. This new ratio of physical-to-logical qubits is close to the theoretical optimum[27].

Instead of 20 million standard superconducting qubits or even 360 thousand cat qubits, 2D LDPC would allow us to run our favorite benchmark algorithm, Shor's algorithm, on an RSA 2048-bit key with only around 100 thousand cats.

THIS IS A SIGNIFICANT 200 TIMES FEWER QUBITS THAN OTHER STATE-OF-THE-ART APPROACHES.

And Shor's algorithm isn't the only algorithm expected to require large numbers of logical qubits and huge numbers of quantum gates. All known practical applications likely to bestow a disruptive business advantage will do so, as well.

The combination of cat qubits and advanced connectivity for ultra-efficient error correction, like LDPC, dramatically reduces the requirements for building a powerful quantum computer. In this image, we show the hardware resources needed to run Shor's algorithm on a 2048-bit key, a 200-fold reduction, which impacts not only the computer's footprint but its energy consumption as well.



Here are just a few examples:

- **Breaking binary elliptic-curve cryptography,** which is used in many cryptocurrencies, including Bitcoin, will require 1,000 to 10,000 logical qubits and 10^{11} quantum gates with a gate error of 10^{-12} [25].

- **Realizing a quantum advantage for financial derivative pricing** will require about 10,000 logical qubits and 10^{10}–10^{11} quantum gates with a gate error of 10^{-13} [28].

- **Simulating nitrogen fixation on nitrogenase,** which is key to the production of most fertilizers, will require 2,196 logical qubits and 10^{11} quantum gates with a gate error of 10^{-13} [6].

IF WE USED A STANDARD APPROACH, EACH HIGH-FIDELITY LOGICAL QUBIT WOULD BE COMPOSED OF THOUSANDS OF PHYSICAL QUBITS. USING CAT QUBITS, WE WILL ONLY NEED 15[27].

Needless to say, massive savings will accumulate.

We think this is the best route to useful quantum computing. Among superconducting circuits, it is the most economical in terms of materials and energy needs, not to mention operational and engineering complexity. Most importantly, it looks like it's going to be the earliest to deploy at scale.

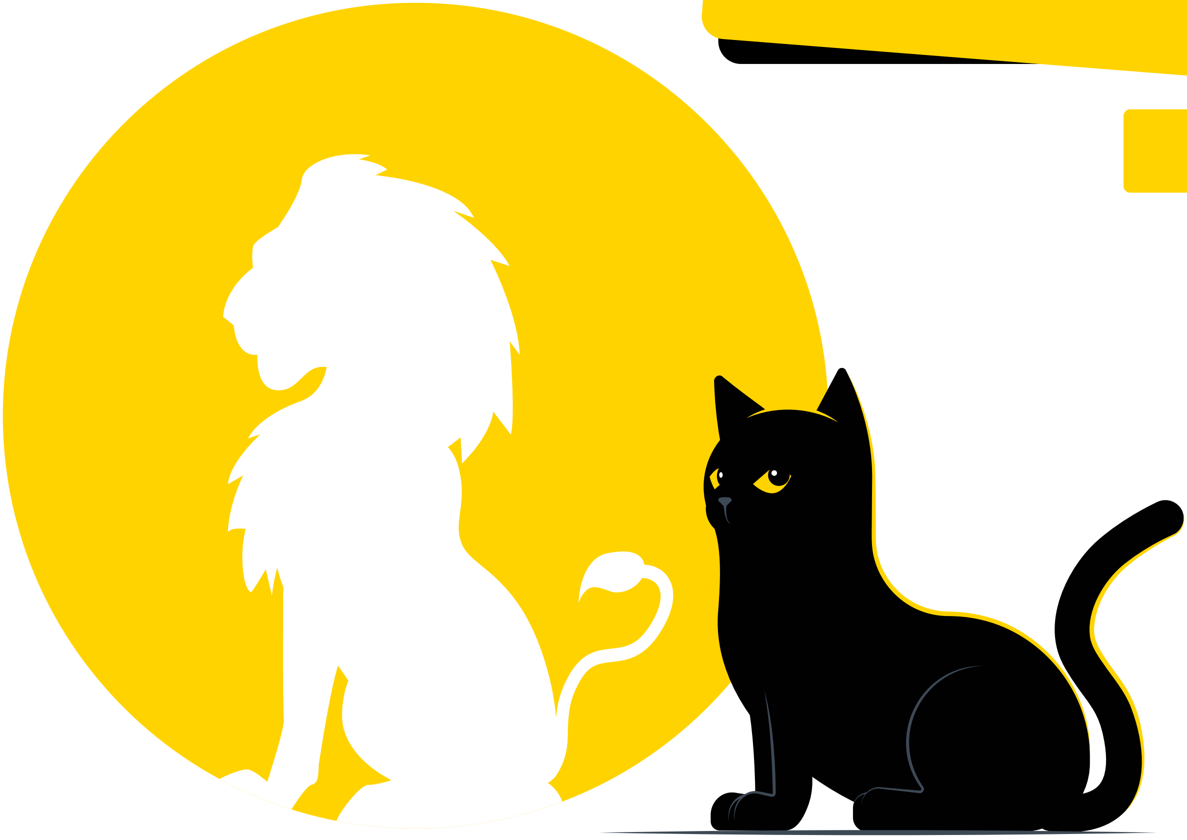
Furthermore, for any given number of qubits assembled and controlled, cats are expected to have drastically higher computing power and reliability versus other platforms while being comparable in terms of footprint, resource needs, and operational complexity.

SIMPLY PUT, A HARDWARE-EFFICIENT ARCHITECTURE ALLOWS US TO ACHIEVE MORE WITH LESS.

This is what distinguishes our approach from others, and you'll see this reflected in our roadmap.

IN A NUTSHELL

Cat qubits simplify quantum error correction by operating in one dimension. This allows the use of more efficient codes, such as LDPC codes with connectivity in only 2 dimensions, which significantly reduces the number of physical qubits required. This is how our cat qubits can offer a faster, more economical path to useful quantum computing.



// FOURTH PART

ALICE & BOB'S QUANTUM COMPUTING ROADMAP

The Path to the First Useful Quantum Computer

We cannot stress enough that a quantum computer which decoheres is not a quantum computer.

From our first steps in 2020, marked by our founders' paper in Nature Physics, to the release of our record-breaking Boson 4 chip on the cloud in May 2024, fighting decoherence has been our core mission. Today, after four years of continuous research, our cat qubit holds the world record among superconducting qubits for bit-flip times, and we're developing our first logical qubit.

Looking ahead, we're committed to launching Graphene in 2030, just one decade after we started. This will be a universal, fault-tolerant quantum computer designed to solve real-world problems. Although we've set milestones along the way, we've deliberately not set dates for these since the timing of research is rarely predictable. We liken this to playing golf: at the end of the round, we won't care if we met our predictions for each hole, but we'll care about wearing a green championship jacket in six years.

To achieve this, our five milestones, each tested and demonstrated on a distinct chip series, will move us closer to a practical quantum computer:

1. Master the Cat Qubit
Our Boson series will establish a reliable, reproducible cat qubit capable of storing quantum information and resisting bit-flip errors.

2. Build a Logical Qubit
Our Helium series will feature our first error-corrected, below-threshold logical qubit.

3. Fault-Tolerant Quantum Computing
Our Lithium series will connect logical qubits and demonstrate the first error-corrected logical gate.

4. Universal Quantum Computing
Our Beryllium series will implement a universal set of logical gates, enabled by magic state factories and live error correction.

5. Useful Quantum Computing
Our Graphene series will feature 100 low-error logical qubits that can be integrated into industrial computing facilities.

By 2030, we're confident that we'll have a computer that users will want to buy. And it won't be because it's quantum, but because it'll give results that won't be achievable in any other way.

Graphene will be the first of tomorrow's useful quantum computers, not the last. Its mission is to redefine the boundaries of computation. Our journey will continue, further scaling fault-tolerant quantum computers while further reducing errors.

Now, let's unfold our detailed research plan for the next five years. ■

IN A NUTSHELL

Since our inception, our mission has been to fight decoherence and develop fault-tolerant quantum computers. Leveraging our cat qubit, our goal is to create a useful quantum computer with 100 high-fidelity logical qubits by 2030. This is set to be the world's first quantum computer with meaningful applications, ushering in the dawn of the quantum era.

INTERACTIVE ROADMAP

// MILESTONE 1

MASTER THE CAT QUBIT

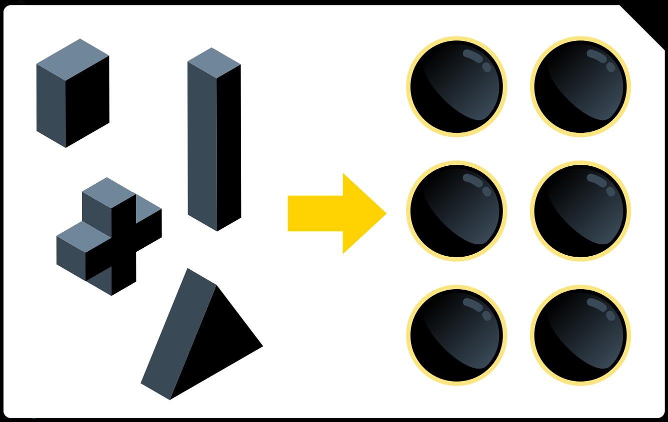
2024 COMPLETE

During the 2010s, cat qubits were experimentally realized only a handful of times, and they didn't demonstrate their bit-flip protection qualities. The first cat qubit with its signature noise-bias wasn't successfully created and controlled until 2020. This seminal work by our founders, published in Nature Physics[23], led to the formation of Alice & Bob.

The first milestone of our journey was to master this new qubit, and we've done that. The release of Boson 4 on the cloud marked a foundational step toward building a quantum computer with cat qubits.

Challenge 1

Herd the Cats



In our formative years, each cat qubit was a unique experiment. We thought of them as handcrafted art pieces, each requiring specialized setups and adjustments. Unfortunately, their lack of reproducibility slowed our innovation.

To scale our quantum technology, we needed to master the reproduction and calibration of our qubits. Achieving this level of mastery demanded improvements in physics, hardware design, and nanofabrication.

By 2024, we had developed a deep understanding of these factors and adopted a standardized innovation process called DeepTech Motion.

Challenge 2

Achieve Record Bit-Flip Protection

Bit-flip protection is a cornerstone of our strategy. This allows cat qubits to efficiently run error correction at scale using far fewer qubits than other architectures. Over the years, our Boson series of single-cat chips has progressively raised the bar in bit-flip resistance and qubit design.

In 2023, our cat qubits were 10,000 times more resistant to bit-flip errors than our first-generation chips, achieving a bit-flip lifetime of 10 seconds. At the same time, we reduced cabling requirements and halved the overall footprint.

The culmination of these efforts was Boson 4, which we launched globally over the cloud in May 2024. This chip set the bit-flip lifetime record for any superconducting qubit: over 7 minutes. This achievement provided definitive proof that our cat qubits operate as theory predicts. We demonstrated that their intrinsic noise-bias makes them ideal for large-scale, hardware-efficient error correction.

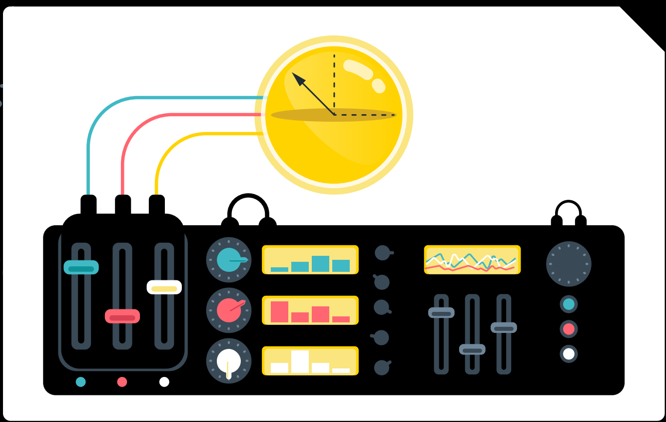
// MILESTONE 1

MASTER THE CAT QUBIT

2024 COMPLETE

Challenge 3

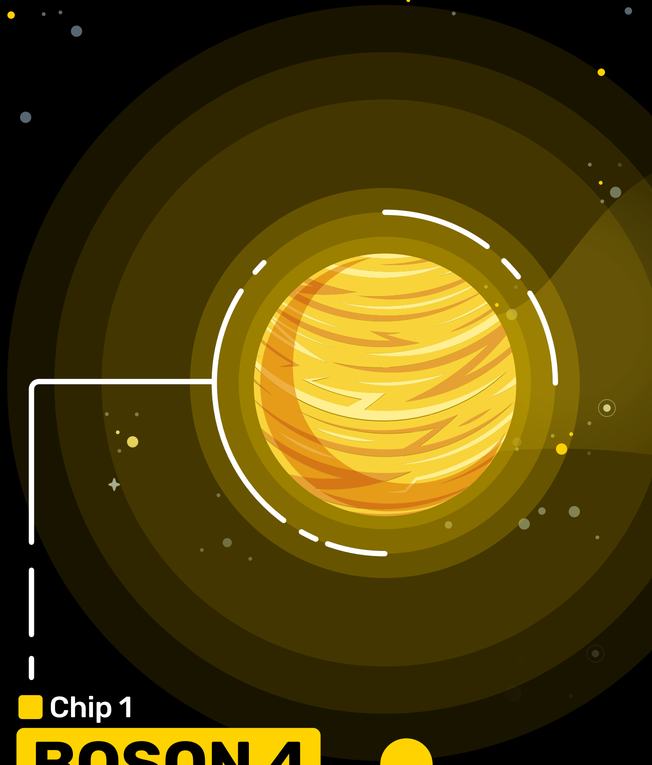
Run Cat Qubit Operations



A single cat qubit, in isolation, can store a single quantum bit of information. To make meaningful use of this information, we need a suite of operations that can manipulate the qubit effectively.

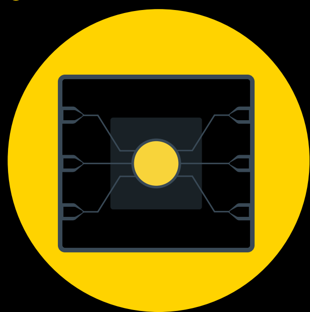
The first critical step was developing elementary single-qubit operations: various state preparations, bit and phase measurements, and fundamental gates like the Z gate (phase gate). These operations aren't sufficient for running full quantum algorithms, but they lay the groundwork for a future universal gate set.

Our objective at this stage was to perform and validate these fundamental manipulations while preserving the cat qubit's inherent noise bias.



Chip 1

BOSON 4



Cat Qubits **1**

Logical Qubits **0**

// MILESTONE 2

BUILD A LOGICAL QUBIT

WE ARE **HERE**

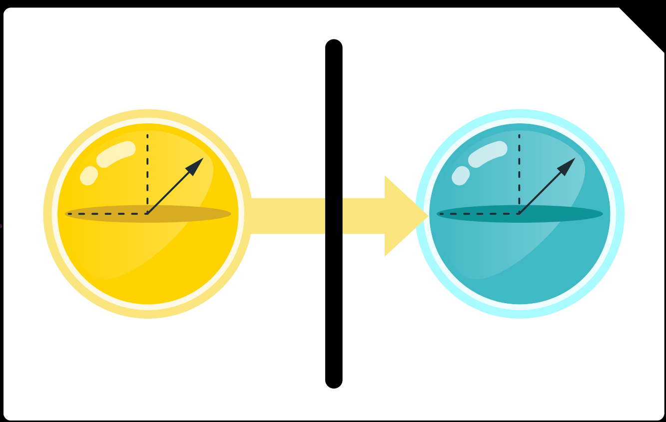
Now that we've mastered a single physical cat qubit, our next critical step is to develop a logical qubit. It must be error-corrected and under threshold. This is where we currently are on our roadmap, making this our primary goal at this time.

If you think about it, we protected our cats against bit-flip errors to reach our first milestone, so we're simply shifting our focus to the remaining error. This means adding a robust error correction process, which involves detecting and correcting phase-flips across a series of cats.

This milestone is necessary for building a fault-tolerant system. It will take us from developing a bias-preserving CNOT gate, through reliably detecting phase-flips, to connecting qubits in lines.

Challenge 1

Mirror the Phase



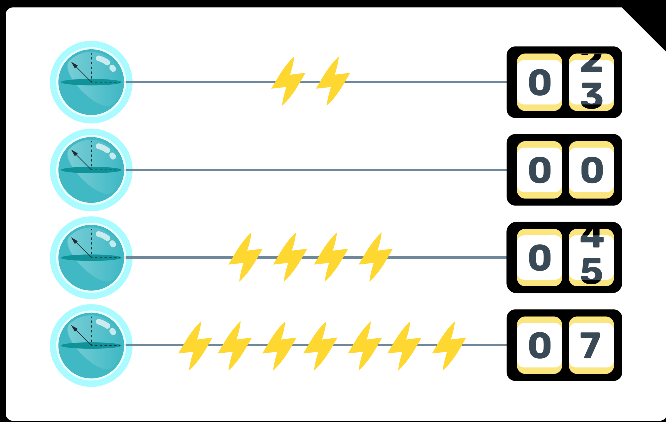
The first step in building a logical, error-corrected qubit is developing the Controlled-NOT gate, often referred to simply as the CNOT. The CNOT is a critical two-qubit operation that can 'mirror' the phase of one cat qubit onto another.

The importance of the CNOT is that it allows us to transfer the phase-flip effects affecting our data qubits to our auxiliary qubits. We can then measure the phases of the auxiliary qubits without corrupting the quantum information carried in the data qubits. The key challenge here is that our CNOT gate must be bias-preserving. In other words, it must not reintroduce bit-flip errors, as that would undermine the cat qubit's noise-bias advantage.

In March 2024, at the meeting of the American Physical Society, we presented promising experimental results from our Hydrogen chip: we showed the first CNOT operation involving two cat qubits. Since then, we have made further improvements to the gate's speed and reliability, which helps to reduce the likelihood of reintroducing bit-flips.

Challenge 2

Count the Flips



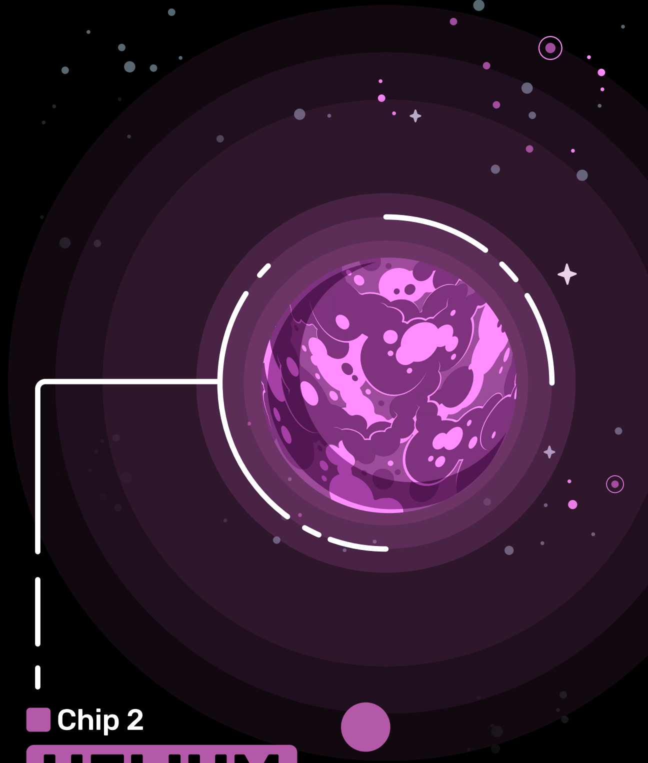
To detect and correct phase-flip errors, our next step will be to create a fast and reliable X measurement protocol. This measurement, also known as the phase measure or Mx, will allow us to interpret the results of CNOT operations. Specifically, we need to determine whether phase-flips have occurred and, if so, where.

The goal of the X measure is to repeatedly monitor the phases of the auxiliary qubits, which accumulate the results of multiple CNOT operations on the data qubits. For this mechanism to be useful on larger devices, such as our Helium chip series, it needs to be precise and fast.

// MILESTONE 2

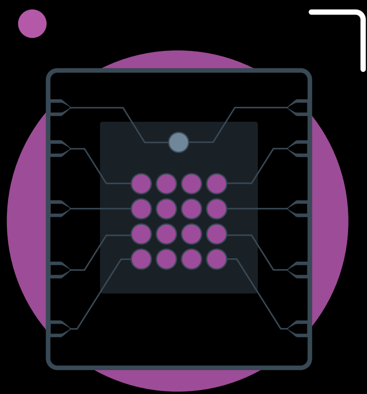
BUILD A LOGICAL QUBIT

WE ARE **HERE**



Chip 2

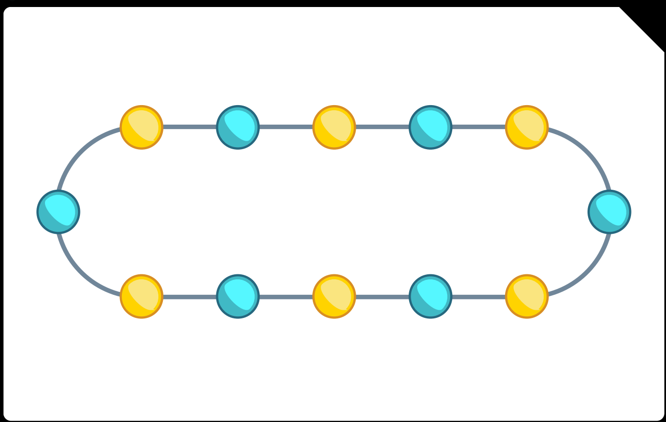
HELIUM



Cat Qubits	16
Logical Qubits	1
Clock Speed (μs)	1.5
Logical Error Rate	10^{-2}

Challenge 3

Link the Qubits



To fully implement quantum error correction, we need to run CNOT gates and phase measurements sequentially across multiple qubits in a chain. This is the cat qubit's signature error correction scheme, the repetition code. This crucial step requires degree-2 connectivity, meaning that each cat qubit must connect to at least two neighbors in a line.

As we create this linear arrangement of cat qubits, we need to ensure that their key features, such as their bit-flip protection and gate efficiency, don't degrade. Each additional connection brings a risk of introducing new noise or increasing errors, which would undermine our main goal: correcting errors.

We'll thoroughly test all the elements necessary for this connectivity on our Hydrogen chip. After that, we'll be able to launch our first under-threshold logical qubit chip, Helium. Hydrogen is a transitional chip series between Boson and Helium, which is why it is not listed as a milestone chip series.

// MILESTONE 3

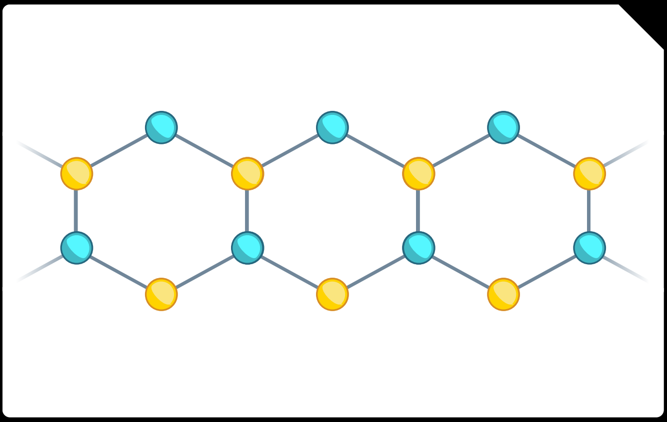
FAULT-TOLERANT QUANTUM COMPUTING

Once we have a single hardware-efficient logical qubit, our next major milestone will focus on scaling up our design to support a network of interacting logical qubits. This means moving to a robust, interconnected grid that enables logical quantum computations.

This third milestone will introduce a series of advances to support this scaling. These include establishing a hexagonal qubit grid and using advanced fabrication techniques to significantly increase qubit density. Equally important, we'll begin developing logical gates, with a focus on the logical CNOT gate as the foundation for universal quantum computing.

After these three challenges are cleared, we will tape out our first multi-logical qubit device, Lithium.

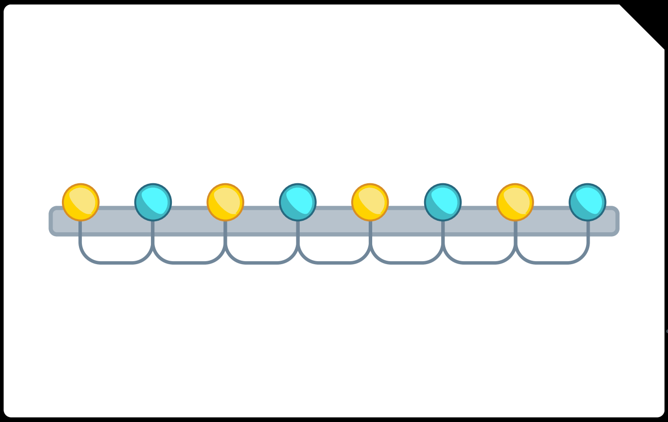
Challenge 1
Connect Logical Qubits



Creating a single, hardware-efficient logical qubit is an essential milestone, but useful quantum computers will require hundreds of interconnected logical qubits. To achieve this, we'll move from a linear, degree-2 connectivity to a complex, hexagonal, degree-3 grid structure. In this configuration, each physical cat qubit will be connected to three neighbors: two within their logical qubit and one from another logical qubit.

Establishing hexagonal connectivity introduces new engineering challenges, as these connections must not introduce additional noise or errors. Achieving this level of connectivity without performance degradation will be fundamental to building our Lithium chip, which will feature multiple logical qubits interacting with one another in a robust network.

Challenge 2
Double the Density



To scale quantum processors efficiently, we must fit significantly more cat qubits in a hexagonal grid layout onto a single chip. Increasing qubit density requires advanced fabrication techniques, notably flip-chip technology. This technology was developed for standard semiconductors but is now being applied to quantum devices as well.

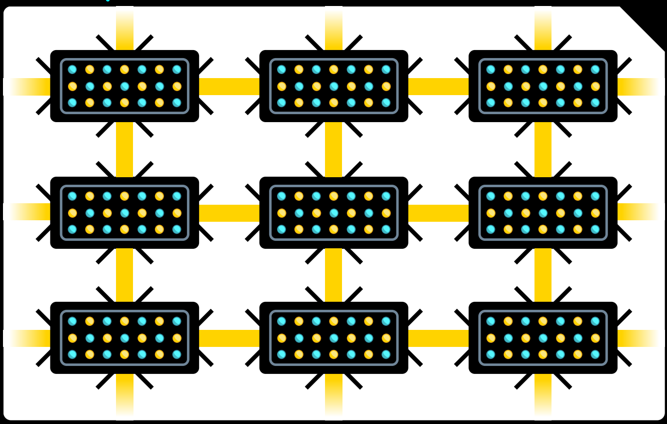
This flip-chip approach allows circuits to utilize both sides of the processor, with qubits packed on one side and their connections printed on the other. This effectively doubles qubit density, while keeping the quantum processing unit's footprint unchanged.

// MILESTONE 3

FAULT-TOLERANT QUANTUM COMPUTING

Challenge 3

Run a First Logical Gate

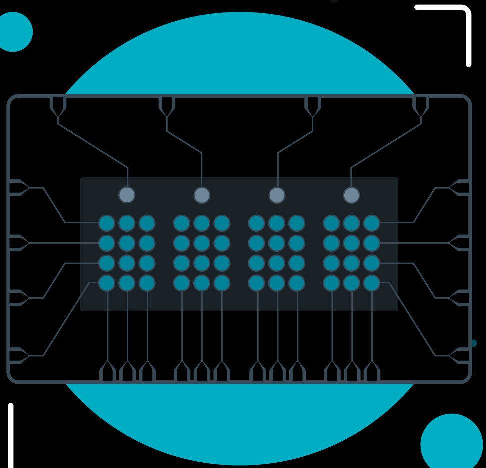


As we transition to working with logical qubits, it will become crucial to develop error-corrected logical gates. These gates must not reintroduce errors into already-corrected qubits. Developing a robust, logical CNOT gate, in particular, will be essential.

The logical CNOT serves as the cornerstone for fault-tolerant quantum operations. It is fundamental to multi-logical-qubit systems, like our upcoming Lithium chip, and will be the first logical gate in our universal gate set. This foundational set of gates forms the basis of all other gates and will enable us to perform universal quantum computations.

Chip 3

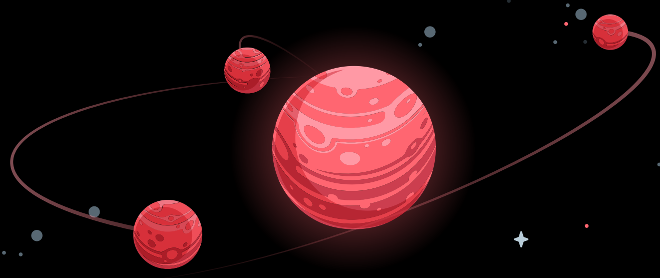
LITHIUM



Cat Qubits	**48**
Logical Qubits	**4**
Clock Speed (µs)	**0.8**
Logical Error Rate	**10^{-3}**

// MILESTONE 4

UNIVERSAL QUANTUM COMPUTING

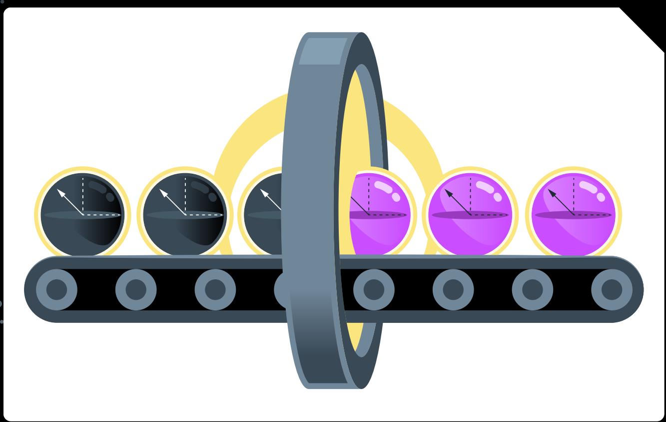


A universal quantum computer is one that is capable of running any quantum algorithm, and a prerequisite for this is having a complete universal gate set. We must complete our set with the Toffoli gate, an advanced operation that requires solving two key challenges: producing a continuous stream of magic states and implementing live error correction. The latter monitors and corrects errors during gate operations. We will also develop our quantum firmware at this stage. The firmware will be a control layer essential for orchestrating the rapid hardware operations needed for our growing architecture.

With these advancements, the Beryllium chip series will be our first processors capable of universal quantum computing. They will lay the groundwork for larger systems designed to solve classically intractable problems.

Challenge 1

Create Magic States

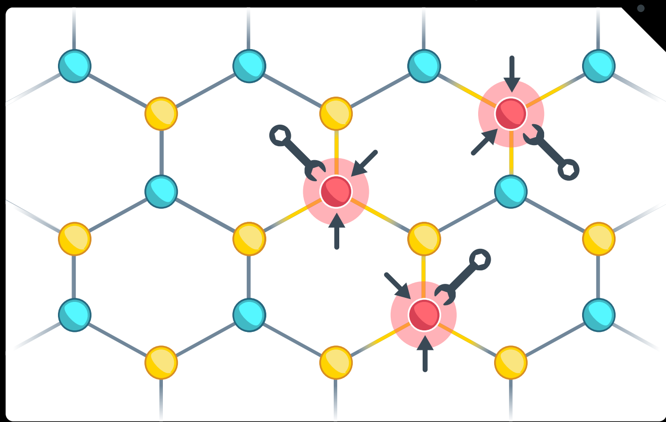


The Toffoli gate is a three-qubit gate that is critical for universal quantum computation. Implementing a logical Toffoli requires a continuous supply of qubits set in what is called a 'magic state.' These magic states get 'created' and 'spent' throughout an algorithm's execution. They are essential to maintaining the gate's logical integrity and preventing it from reintroducing errors into the system.

Creating a steady supply of these states, though, is both technically demanding and resource-intensive. It requires that we have specialized areas on our chip, which we designate as 'magic state factories.'

Challenge 2

Correct Errors, Live



To implement gates like the Toffoli gate, our error correction process must be able to detect and address errors while our quantum algorithms are running.

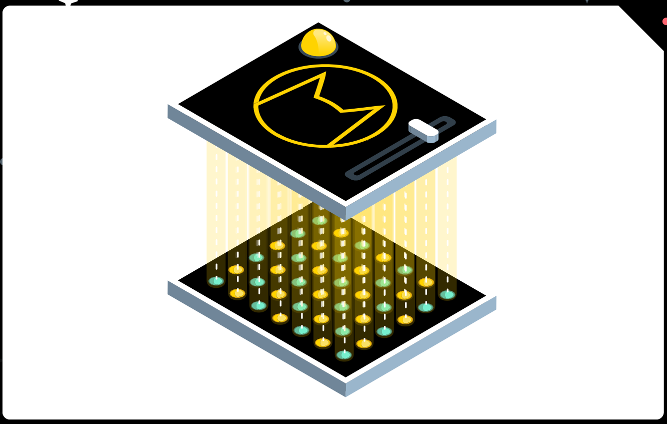
To meet this challenge, we need to develop a fast and efficient 'live decoder.' This specialized classical computing functionality will allow us to correct errors rapidly enough to avoid slowing down operations. The live decoder must match the speed of the gate operations, ensuring that error correction remains synchronized with gate execution and doesn't introduce delays that could compromise the computation.

// MILESTONE 4

UNIVERSAL QUANTUM COMPUTING

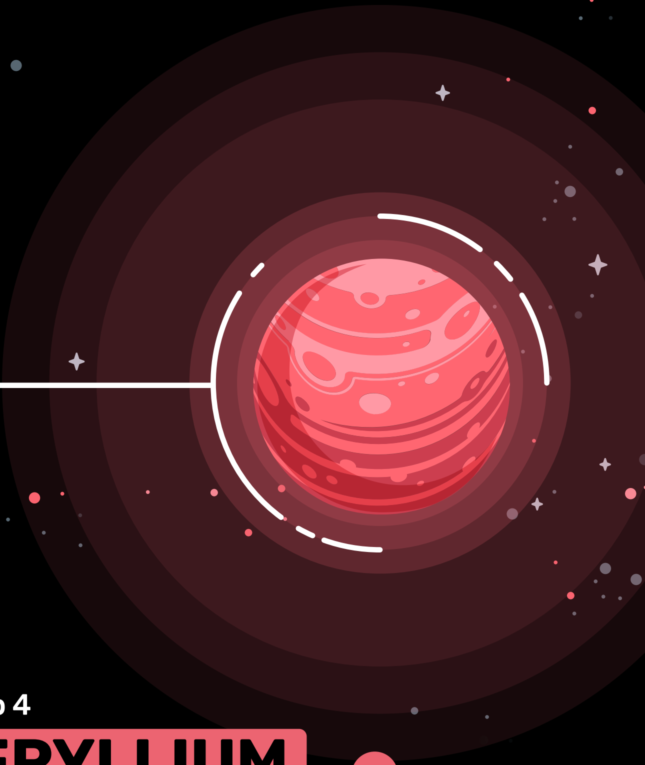
Challenge 3

Build the Quantum Firmware



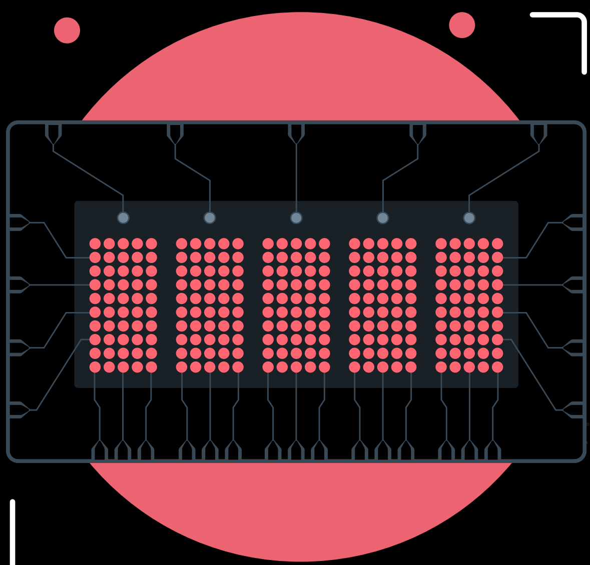
As our chips get larger and our gates grow in complexity, orchestrating the multitude of operations required at the hardware level becomes an increasingly formidable challenge. Each step in quantum computation depends on precise, synchronized pulses of microwave photons, which govern qubit behavior.

This is where the quantum firmware comes into play. You can think of it as a specialized operating system for our QPU. Firmware translates high-level algorithmic instructions into the exact control signals that each qubit responds to. This ensures that all operations are executed with the required timing and precision.



Chip 4

BERYLLIUM

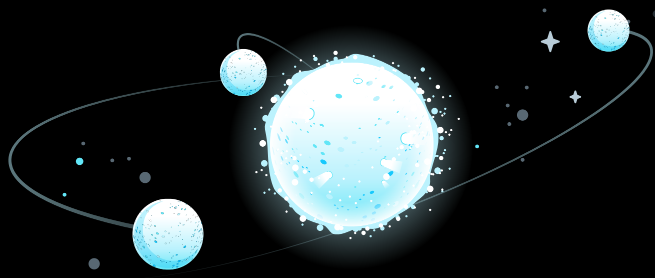


Cat Qubits	250
Logical Qubits	5
Clock Speed (μs)	0.8
Logical Error Rate	10^{-4}

// MILESTONE 5

USEFUL QUANTUM COMPUTING

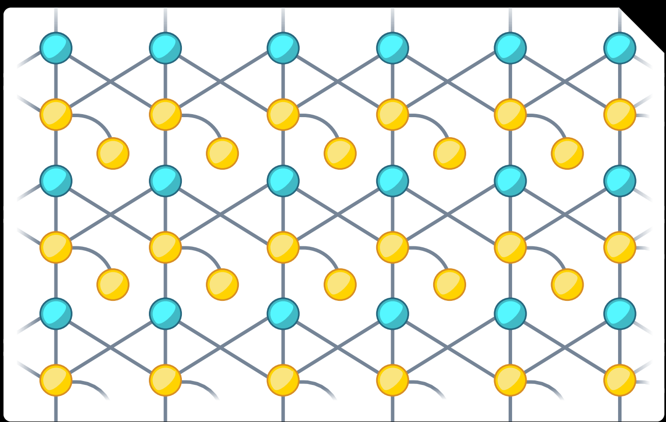
2030 START OF THE QUANTUM ERA



Estimates vary, but somewhere around 100 high-fidelity, logical qubits, we'll finally unlock some truly transformative applications for quantum computing. We'll be able to tackle problems that no classical computer can solve, delivering real-world value in fields like fundamental research. However, reaching this scale will require pushing hardware efficiency and quantum engineering to their limits.

In this phase, we will focus on maximizing the capability of every qubit and optimizing our hardware infrastructure. These engineering breakthroughs will form the foundation of our ultimate goal: building a universal, fault-tolerant quantum computer that demonstrates practical quantum advantage. It will be named Graphene, and we'll deliver it before the end of this decade.

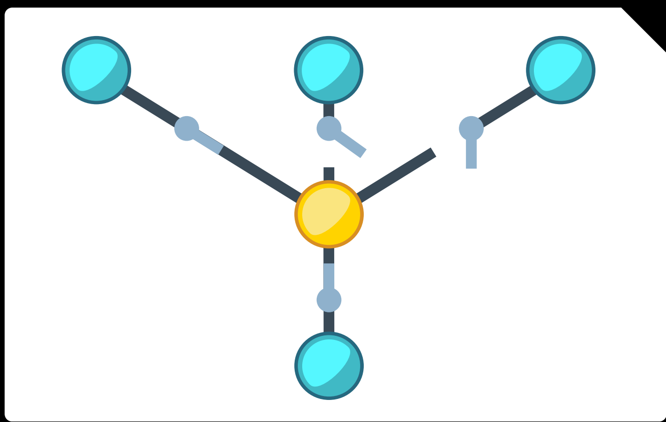
Challenge 1
Top-up Hardware Efficiency



Our goal is to reach the mathematical limits of hardware efficiency, and the connectivity degree required for LDPC (Low-Density Parity-Check) is key to achieving it. This advanced layout will connect each cat qubit with up to five neighbors, enabling reliable error correction with far fewer physical qubits than our already-efficient repetition code. Using Shor's algorithm as an example, we would need 3.5X fewer qubits, compared to our previous architecture, to factor a 2048-bit integer.

LDPC codes work by allowing physical qubits to contribute to more than one logical qubit simultaneously, thus, minimizing hardware use. This approach achieves comparable performance to other qubit technologies but, with cat qubits, uses up to 200 times fewer qubits[27].

Challenge 2
Switch Qubits



LDPC codes require selective connectivity, meaning specific qubits must interact under certain conditions with certain qubits without interfering with others. This is where tunable couplers come into play. Tunable couplers act as dynamic 'switches' that can connect and disconnect qubits on demand allowing intended interactions while preventing unwanted ones.

In our cat qubit architecture, tunable couplers must be developed with particular attention to preserving the qubits' inherent noise-bias properties. Additionally, by keeping qubit connections 'off' when their respective qubits aren't involved in a calculation, tunable couplers can reduce the potential for errors and optimize resource usage on the chip.

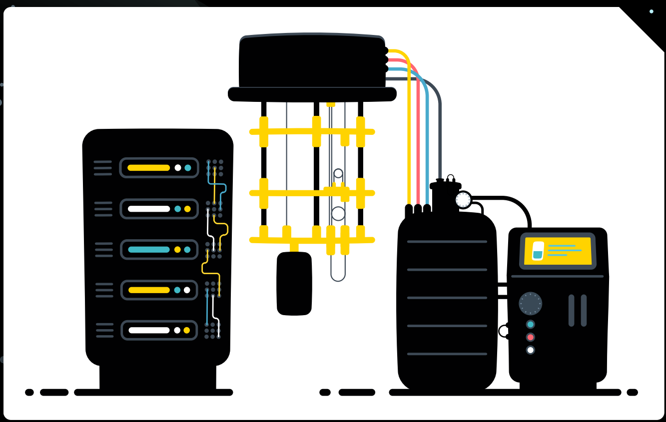
// MILESTONE 5

USEFUL QUANTUM COMPUTING

2030 START OF THE QUANTUM ERA

Challenge 3

Improve Enabling Technologies

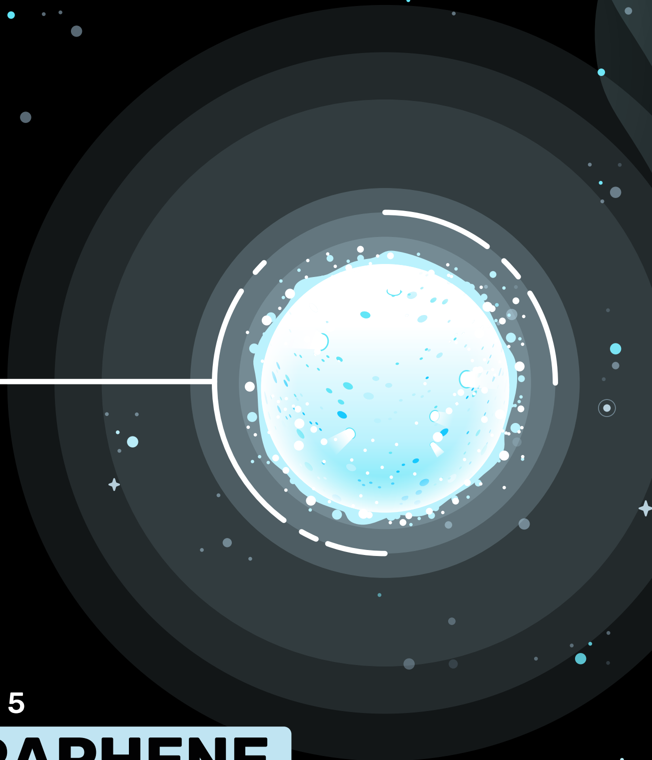


As we produce larger quantum processors, the supporting infrastructure must evolve too. It must become far more efficient in terms of footprint, energy consumption, and operational complexity.

Quantum systems require incredibly complex environments to function: extensive cabling, electronics, classical computing resources for error correction, and fridges that reach temperatures colder than outer space. All of these resources scale with the number of qubits.

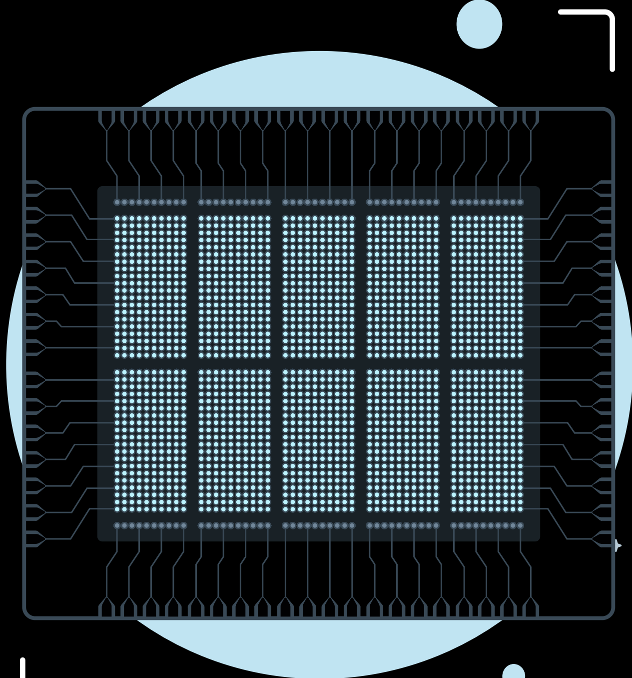
Luckily for us, cat qubits need far fewer physical qubits per logical qubit. This reduces the demand for infrastructure compared to traditional approaches.

Many industry players across the quantum hardware stack are actively contributing to this effort by scaling and refining cabling, microwave delivery systems, and cryogenic engineering. We will continue partnering with experts across these areas to integrate these enabling technologies into our first universal, fault-tolerant quantum computer. And by 2030, Graphene will be solving real-world use cases.



Chip 5

GRAPHENE



Cat Qubits	2000
Logical Qubits	100
Clock Speed (μs)	1
Logical Error Rate	10^{-6}

What's after that?

With 100 logical qubits, we will confidently step beyond classical computing's limits in simulating quantum reality. We'll escape both brute-force simulations and ad-hoc solutions, and with accuracy. Graphene will certainly thrill researchers in a variety of fields, but this accuracy will also allow us to solve problems with commercial value. This is the critical distinction between 100 logical qubits and 100 physical qubits.

Graphene will prove the feasibility of cat qubits as the preferred hardware for building universal fault-tolerant quantum computers. We will use our knowledge to continue building bigger and better quantum computers.

These next-generation devices will feature more logical qubits, lower error rates, and faster speeds, unlocking new applications for quantum computers. We'll have already provided a powerful new tool for fundamental science, material science, and chemistry, and we'll be looking toward applications in optimization and machine learning. We'll be opening up use cases in financial services, supply chain operations, and more.

Conclusion:

WHY DO WE DO ALL OF THIS?

WE'RE DOING WHAT WE'VE ALL BEEN TRAINED TO DO AS SCIENTISTS AND ENGINEERS: WE'RE CREATING SCIENCE APPLICATIONS.

HOWEVER, THAT'S JUST NOT THE CASE.

We're driven to build a useful quantum computer because the world needs to solve problems that are fundamentally beyond the reach of the most powerful classical computers we can build.

We live in a world where existential challenges demand ambitious technological solutions. What will be the role of humans in a future where another form of intelligence exists? How can we push the boundaries of drug discovery to combat disease? How can we fix the damage our society is causing to the ecosystem? How can we mitigate the risk of pandemics in an ever more interconnected and globalized world?

These are just a few examples. As engineers, physicists, and technologists, our contribution toward solving these questions is the responsible advancement of science and technology.

We're building a quantum computer for the countless curious minds that will undoubtedly follow us. They will need to answer these questions, plus many other hard questions we can't even predict yet. Nonetheless, we can provide them with the tools they'll need to find the answers.

We and our colleagues across the industry are heirs to Alan Turing and John Von Neumann, the fathers of the classical computer. Without their foundational research, we wouldn't be here today, exploring how to push computing beyond its current limits. Similarly, we aspire to become the pioneers of a new computing paradigm by being the first to build a truly useful quantum computer, the basis upon which generations of other innovators will build.

We're doing this even though no one yet knows the limits of this novel technology. All we know is that a quantum computer will unlock a whole new world of possibilities, both known and unknown.

Together, we have a grand journey ahead; where would you like to go?

THE END

Bibliography

[1] C. Monroe, D. M. Meekhof, B. E. King, W. M. Itano, and D. J. Wineland. "Demonstration of a Fundamental Quantum Logic Gate." Physical Review Letters 75, 4714 (December 1995). https://journals.aps.org/prl/pdf/10.1103/PhysRevLett.75.4714

[2] Nakamura, Y., Pashkin, Y., and J. Tsai. "Coherent Control of Macroscopic Quantum States in a Single-Cooper-Pair Box." Nature 398 (1999): 786–788. https://doi.org/10.1038/19718.

[3] "IBM Makes Quantum Computing Available on IBM Cloud to Accelerate Innovation." IBM UK Newsroom, May 4, 2016. https://uk.newsroom.ibm.com/2016-May-04-IBM-Makes-Quantum-Computing-Available-on-IBM-Cloud-to-Accelerate-Innovation.

[4] Arute, Frank, Kunal Arya, Ryan Babbush, et al. "Quantum Supremacy Using a Programmable Superconducting Processor." Nature 574 (2019): 505–510. https://doi.org/10.1038/s41586-019-1666-5.

[5] Lee, Joonho, Dominic W. Berry, Craig Gidney, William J. Huggins, Jarrod R. McClean, Nathan Wiebe, and Ryan Babbush. "Even More Efficient Quantum Computations of Chemistry Through Tensor Hypercontraction." PRX Quantum 2, no. 3 (July 2021): 030305. https://doi.org/10.1103/PRXQuantum.2.030305

[6] Bobier, Jean-François, Matt Langione, Cassia Naudet-Baulieu, Zheng Cui, and Eitoku Watanabe. "The Long-Term Forecast for Quantum Computing Still Looks Bright." BCG Global, July 18, 2024. https://www.bcg.com/publications/2024/long-term-forecast-for-quantum-computing-still-looks-bright.

[7] Bogobowicz, Michael, Kamalika Dutta, Martina Gschwendtner, Anna Heid, Mena Issler, Niko Mohr, Henning Soller, Rodney Zemmel, and Alex Zhang. "Steady Progress in Approaching the Quantum Advantage." McKinsey Digital, April 24, 2024. https://www.mckinsey.com/capabilities/mckinsey-digital/our-insights/steady-progress-in-approaching-the-quantum-advantage.

[8] "Catalyst Market Size, Share & Trends Analysis Report By Raw Material (Chemical Compounds), By Product by Application (Heterogeneous, Homogeneous), By Region, And Segment Forecasts, 2024 - 2030." Catalyst Market Size & Trends Analysis Report, 2023. Accessed November 13, 2024. https://www.grandviewresearch.com/industry-analysis/catalyst-market.

[9] Elenewski, Justin E., Christina M. Camara, and Amir Kalev. "Prospects for NMR Spectral Prediction on Fault-Tolerant Quantum Computers." 2024. arXiv. https://arxiv.org/abs/2406.09340.

[10] Sood, Neeraj, Tiffany Shih, Karen Van Nuys, and Dana Goldman. "Flow of Money through the Pharmaceutical Distribution System." USC Schaeffer, June 6, 2017. https://healthpolicy.usc.edu/research/flow-of-money-through-the-pharmaceutical-distribution-system/.

[11] Monroe, C., Meekhof, D. M., King, B. E., Itano, W. M., and Wineland, D. J. "Demonstration of a Fundamental Quantum Logic Gate." Physical Review Letters 75, no. 25 (December 1995): 4714-4717. https://doi.org/10.1103/PhysRevLett.75.4714.

[12] Scholten, Travis L., Carl J. Williams, Dustin Moody, Michele Mosca, William Hurley, William J. Zeng, Matthias Troyer, and Jay M. Gambetta. "Assessing the Benefits and Risks of Quantum Computers." Last modified 2024. arXiv. https://arxiv.org/abs/2401.16317.

[13] Bluvstein, Dolev, Simon J. Evered, Alexandra A. Geim, et al. "Logical Quantum Processor Based on Reconfigurable Atom Arrays." Nature 626 (2024): 58–65. https://doi.org/10.1038/s41586-023-06927-3.

[14] da Silva, M. P., C. Ryan-Anderson, J. M. Bello-Rivas, A. Chernoguzov, J. M. Dreiling, C. Foltz, F. Frachon, J. P. Gaebler, T. M. Gatterman, L. Grans-Samuelsson, D. Hayes, N. Hewitt, J. Johansen, D. Lucchetti, M. Mills, S. A. Moses, B. Neyenhuis, A. Paz, J. Pino, P. Siegfried, J. Strabley, A. Sundaram, D. Tom, S. J. Wernli, M. Zanner, R. P. Stutz, and K. M. Svore. 2024. "Demonstration of Logical Qubits and Repeated Error Correction with Better-than-Physical Error Rates." arXiv. https://arxiv.org/abs/2404.02280.

[15] Putterman, Harald, et al. 2024. "Hardware-efficient Quantum Error Correction Using Concatenated Bosonic Qubits." arXiv. https://arxiv.org/abs/2409.13025.

[16] Acharya, Rajiv, et al. «Quantum Error Correction Below the Surface Code Threshold.» arXiv preprint arXiv:2408.13687 (2024).

[17] Aharonov, Dorit, and Michael Ben-Or. «Fault-Tolerant Quantum Computation With Constant Error Rate.» arXiv preprint quant-ph/9906129 (1999). https://arxiv.org/abs/quant-ph/9906129.

[18] Kitaev, A. Yu. «Quantum Computations: Algorithms and Error Correction.» Russian Mathematical Surveys 52, no. 6 (1997): 1191. https://dx.doi.org/10.1070/RM1997v052n06ABEH002155.

[19] Knill, Emanuel, Raymond Laflamme, and Wojciech H. Zurek. "Resilient Quantum Computation." Science 279, no. 5349 (1998): 342-345. https://doi.org/10.1126/science.279.5349.342.

[20] Aliferis, Panos, Daniel Gottesman, and John Preskill. "Quantum Accuracy Threshold for Concatenated Distance-3 Codes." 2005. arXiv:quant-ph/0504218. https://arxiv.org/abs/quant-ph/0504218.

[21] Riverlane. "Quantum Error Correction Report 2024." Riverlane. Accessed November 13, 2024. https://www.riverlane.com/quantum-error-correction-report-2024.

[22] Gidney, Craig, and Martin Ekerå. "How to Factor 2048 Bit RSA Integers in 8 Hours Using 20 Million Noisy Qubits." Quantum 5 (April 2021): 433. https://doi.org/10.22331/q-2021-04-15-433.

[23] Lescanne, R., M. Villiers, T. Peronnin, et al. "Exponential Suppression of Bit-Flips in a Qubit Encoded in an Oscillator." Nature Physics 16 (2020): 509–513. https://doi.org/10.1038/s41567-020-0824-x.

[24] Guillaud, Jérémie, and Mazyar Mirrahimi. "Repetition Cat Qubits for Fault-Tolerant Quantum Computation." Physical Review X 9, no. 4 (December 2019): 041053. https://doi.org/10.1103/PhysRevX.9.041053.

[25] Gouzien, Élie, Diego Ruiz, Francois-Marie Le Régent, Jérémie Guillaud, and Nicolas Sangouard. "Performance Analysis of a Repetition Cat Code Architecture: Computing 256-bit Elliptic Curve Logarithm in 9 Hours with 126,133 Cat Qubits." Physical Review Letters 131, no. 4 (July 2023). https://doi.org/10.1103/physrevlett.131.040602

[26] Bravyi, S., Cross, A.W., Gambetta, J.M. et al. "High-threshold and low-overhead fault-tolerant quantum memory". Nature 627, 778–782 (2024). https://doi.org/10.1038/s41586-024-07107-7

[27] Diego Ruiz, Jérémie Guillaud, Anthony Leverrier, Mazyar Mirrahimi, Christophe Vuillot. "LDPC-cat codes for low-overhead quantum computing in 2D" arXiv:2401.09541, https://arxiv.org/abs/2401.09541

[28] Chakrabarti, Shouvanik, Rajiv Krishnakumar, Guglielmo Mazzola, Nikitas Stamatopoulos, Stefan Woerner, and William J. Zeng. 2021. "A Threshold for Quantum Advantage in Derivative Pricing." Quantum 5: 463. https://doi.org/10.22331/q-2021-06-01-463.

ALICE & BOB

CONTACT

niccolo.coppola@alice-bob.com

WE THANK

Théau Peronnin, Chloé Poisbeau, Paul Magnard, Olivier Ezratty, Cécile Perrault, Jeremy Stevens and the Alice & Bob team.

LINKEDIN

X

YOUTUBE

Our interactive roadmap, available on Alice & Bob's website was developed by:
Karim Fares, Elias Chantiri, Najm Choueiry, Elio Najem.