Jumpstarting Quantum Computing in the Middle and High-School Classroom

Journal:	*Inroads*
Manuscript ID	INROADS-202505-001
Manuscript Type:	Standard Article
General Terms:	Algorithms
Keywords:	quantum computing, high-school, middle school

Jumpstarting Quantum Computing in the Middle and High-School Classroom: A Guide for Teachers and Learners

Experiential Learning

Catching a frisbee is not easy but dogs and Computer Science (CSCI) sophomores seem to be good at it. How they actually do it is still very much subject for debate [6, 7]. That they might be calculating trajectories in real time, using Newton's equations, remains a very unlikely hypothesis. And yet it is undeniable that catching a frisbee demonstrates a working knowledge of physics. How is this knowledge acquired? Recently the concept of "embodied heuristics" [6] has been proposed as a possible operational explanation. Such a heuristic is a distillation of evolved sensory and motor abilities and is the result of practice and evolution. If you want to be good at catching a frisbee you need to practice. And if you keep at it, you get good at it. Catching a frisbee is classical physics. Classical physics is all around us. Interaction with it is inevitable, ubiquitous, vital and fun. Quantum physics, however, is a different story. We cannot directly interact with quantum systems as with classical, so we must build intuition with them in different ways.

Building an Intuition

As Dirac taught us, there is a minimum disturbance that accompanies a measurement, a disturbance that is inherent in the nature of things and can never be improved by experimental technique. "If the disturbance is negligible, then the object is large in an absolute sense, and it can be described by classical physics. [I]f the minimum disturbance accompanying a measurement is nonnegligible, then the object is absolutely small, and its properties fall in the realm of quantum mechanics. The quantum properties of absolutely small particles are not strange; they are just unfamiliar and not subject to our classical intuition." [4] Thus, it may be accurate to say that a quantum object is produced as a particle, propagates as a wave and is detected as a particle with the probability distribution of a wave but what difference does that make to a computer scientist? Why do we need quantum mechanics?

Quantum Computing

At the foundation of our field we have two rewriting systems equivalent in computational power: Turing machines and lambda calculus. Turing machines are important for many reasons, but especially because of two long-held beliefs regarding computation: first, the Church-Turing thesis says that everything that is computable can be computed with a Turing machine, although it could sometimes take a very long time. This correctly suggests there are problems that cannot be computed—they are called undecidable problems, the most famous being the halting problem. Aside from such uncomputable problems, everything else can be computed using a Turing machine.

The extended Church-Turing thesis is the other foundational principle of computer science that says that the performance of all computers is only polynomially faster than a probabilistic Turing machine. In 1993, Bernstein and Vazirani showed that quantum computers could violate the extended Church-Turing thesis. Their quantum algorithm offered an exponential speedup over any classical algorithm for a certain computational task called recursive Fourier sampling. Quantum computation is the only model of computation to date to violate the extended Church-Turing thesis, and therefore only quantum computers are capable of exponential speedups over classical computers. This is because quantum computers harness quantum mechanics to compute by different rules than classical computers do. Although quantum computers don't actually compute faster than classical ones, the rules unique to quantum computing allow algorithms new, shorter routes to solutions.

It's important to understand that quantum computers would not violate the regular Church-Turing thesis. What is impossible to compute will remain impossible. The hope is that quantum computers will efficiently solve problems that are inefficient on classical computers. One such problem is the factoring of very large numbers. Another one is simulating nature with computers. Nature appears to follow the laws of quantum mechanics. Sometimes classical computers can struggle to crunch the numbers to figure out what nature is doing, but quantum computers play by different rules. They don't need to crunch these numbers per se; they can simply mimic nature rather than approximate it numerically like the classical computers do—and that's because, like nature, quantum computers are quantum. The potential here is enormous not just for understanding physics but for designing new materials, and medicines, for instance.

Student Agency

Having decided that the topic is important we now ask ourselves what learner-sighted teaching technique is best suited here. Education should foster independent exploration and construction of knowledge, rather than passive acceptance of instruction, encouraging agency in students. Though we agree that a motivated student will always be in pursuit of knowledge, in school we often find knowledge to be in heavy pursuit of the student. Furthermore, we believe all students are intrinsically motivated to learn but become unmotivated if they repeatedly fail. Every student has the basic needs to belong, to be competent and to influence what happens to them; motivation to learn only exists when these three conditions are satisfied[1]. With this in mind we have developed an operational approach to jumpstarting quantum computing education in learners as early as middle or high school. Here we restrict ourselves to present the phase kickback phenomenon and the Bernstein-Vazirani algorithm using just the basic rules of arithmetic. Our approach is based on a string-rewriting system invented by Terry Rudolph and introduced in his 2017 book "Q is for Quantum" [10, 11, 12]. We start from classical bits and little by little we introduce phase, superposition, and interference. We show the simple rules that can help a middle school student trace qubits through a quantum circuit. We show how to verify what we do, using the misty states formalism, with circuits implemented in Qiskit. The reader is invited to read along with a pen and some paper. A laptop would come in handy as well.

[1] For all the talk that dogs and CSCI sophomores have a short-attention span, the truth is that individuals in both groups typically exhibit laser focus when the motivation (and determination on what to focus) comes from inside.

Misty States

The 12-year-olds of today may well have access to large quantum computers before they leave their teenage years. Yet a standard educational trajectory would see them still several years away from learning enough quantum theory to explore this technology's amazing potential meaningfully. In addition to barriers of convention ("This is the order in which things have always been taught") there are math-related barriers ("You can't understand quantum theory until you have mastered linear algebra in a complex vector space"). But, as has been shown, and in true CSCI spirit, it is possible to replace linear algebra with some string-rewriting rules [10] which are no more complicated than the basic rules of arithmetic. These rules are very simple indeed but we have to warn the reader of underestimating them. In class we emphasize that mastery of any system, no matter how simple, requires both attention and especially practice. When these two conditions are satisfied we're convinced that the learner will be very succesful.

Also important is to note that our focus is quantum computing (QC) and not quantum mechanics (QM) or quantum physics in general. Learning QC is much easier [3] than learning QM because QC deals with a simple subset of QM, as follows: (a) a qubit—the foundation of quantum computing—is the simplest non-trivial quantum system; (b) you never have to solve the Schrödinger equation, or even learn what it is, because the quantum systems that carry out quantum computations evolve in a controlled manner based on the quantum gates applied to them; and (c) there's already a model of quantum computation, so the most difficult aspect of quantum mechanics—the art of applying it to real systems—is absent. We approach presentation from the mindset of maker-centered learning: "What I cannot create I cannot understand" is a good description of that persuasion and a quote from Richard Feynman. From here on, in our discussion of any QC concept, we advocate an environment of concrete representations via Python, Qiskit and the misty states formalism (the method developed and introduced by Terry Rudolph).

Maker-Centered Learning

According to Piaget "children in the early years of primary school need concrete[2] objects, pictures, actions, and symbols to develop mathematical meanings." The same is true of students who lack a certain background or affinity for the pure structures of mathematics. This is where the simplicity of the misty state formalism shines through. Piaget also said "[l]ogic and mathematics are nothing but specialized linguistic structures." The misty state formalism can facilitate access to both. Another quote, from Seymour Papert, is relevant here: "My basic idea is that programming is the most powerful medium of developing the sophisticated and rigorous thinking needed for mathematics." So our approach is trying to scaffold the knowledge needed to understand quantum computing and quantum information science starting from computing in Python in a notebook (Google Colab). We build an understanding of the misty state formalism and then use it to define, recognize and synthesize (operationally, in Qiskit and Python) the following concepts: superposition, phase, interference, entanglement, quantum gates and quantum circuits, the Deutsch-Josza algorithm, the Grover search algorithm, the Bernstein-Vazirani algorithm (and the phase kickback phenomenon that

[2] Our brains need to interact with something in order to create a model of it. As Papert puts it: "You can't think about thinking without thinking about thinking about something."

makes it possible) along with superdense coding and the GHZ game (quantum pseudo-telepathy via quantum entanglement). We then need to extend the system and present quantum teleportation and the phenomenon known as entanglement swapping (i.e., teleportation of entanglement). In this paper we only present the Bernstein-Vazirani algorithm via phase kickback and misty states. The rest has been presented and is available elsewhere and is now essentially part of the CS2023 report as a separate knowledge unit (KU).

Quantum Flytrap

We emphasize that our main goal is not quantum mechanics. But we need to stress that "[s]tudents and professionals interested in quantum information sciences need to adopt a different way of thinking than the one used to construct today's (classical) algorithms. This certainly presents tremendous challenges, since, for many years, computer science students have been led to believe that they can get by with some knowledge of discrete mathematics and little understanding of physics at all. [However, of necessity, in quantum computing w]e are going back to the age when a strong[er] relationship between physics and computer science existed." [8] Having said all of this we also need to point out that we don't consider *detailed* knowledge of QM a necessity for a CSCI student unless they decide to choose a career in building quantum computing hardware. Here (as is also recommended in the CS2023 KU) we only promote an appreciation of (and familiarity with) the main quantum concepts: qubit, state, phase, interference, entanglement, teleportation, measurement, sensing, coherence, quantum communication and the main differences between QIS and QM. An environment facilitating direct interaction with these concepts is the Quantum Flytrap [2] which self-describes as a no-code IDE for quantum computing. We strongly encourage its use in the classroom and labs. We made it clear that we consider experiential learning a *sine qua non* feature of learning for the kind of learners and topics we have in mind. In such a process building an intuition via embodied heuristics is fundamental but direct interaction with the world of the very small is (a) expensive and (b) has to be mitigated since we're so big. John Preskill once remarked: "Perhaps kids who grow up playing quantum games will acquire a visceral understanding of quantum phenomena that our generation lacks." With this in mind we advocate the use of Quantum Flytrap as a tool to complement the system developed and introduced by Terry Rudolph (the misty states formalism) which we proceed to introduce next.

Bernstein-Vazirani

The problem states that we have a circuit in which we have placed a number of (quantum) gates. The circuit will be presented to us as a black box. It will have a number of inputs and an equal number of outputs. We will be asked to determine the internal connectivity of the black box by just interacting with it from the outside. Using only classical physics (gates and principles) we can only conclude that the task of determining what the black box looks on the inside is linear in the number of inputs. But if we are allowed to use quantum physics (both hardware and principles) the same task can be solved in just one step, regardless of how many inputs the circuit has. Here's an example:

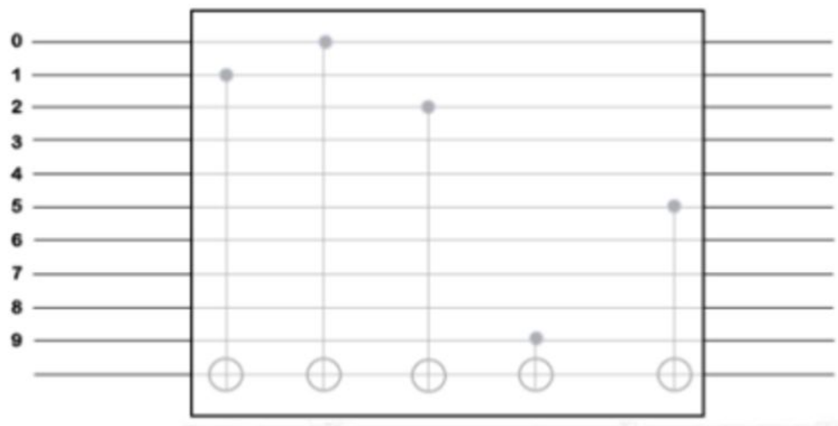


Figure 1. A Bernstein-Vazirani challenge. The black box contains C-NOT gates; they are introduced below.

We now proceed to define the gates and the formalism we need. Familiarity with the first part of [10] is desirable but won't be assumed. As a result we first introduce some of the material already in the book (NOT and C-NOT gates along with the Hadamard (PETE) gate). We then proceed to prove the phase kickback phenomenon via misty states and use it to solve the Bernstein-Vazirani challenge.

The NOT Gate

An excellent resource here is the 20-minute video [12] available on the book's [11] website.

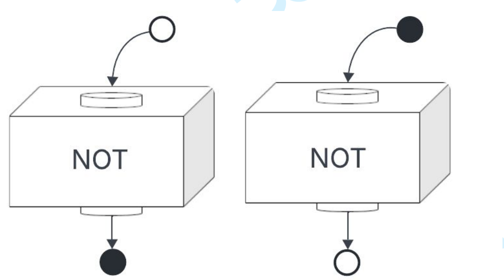


Figure 2. The NOT gate flips its input: NOT(W) = B and NOT(B) = W.

The classical bits are 0 and 1. We can represent them as W and B and draw them as a white or black ball. Indeed they are classical values. A quantum bit (qubit) is a more complex entity but when we measure a qubit we only get one of these two values, W or B. The effect of this gate is consistent with our knowledge of the classical NOT gate.

The C-NOT Gate

The controlled-NOT (C-NOT) gate has two inputs: a target and a control. It works by flipping the target, i.e., acting as a NOT gate on the target when the control is a black ball. Here's a diagram, redrawn from the book:

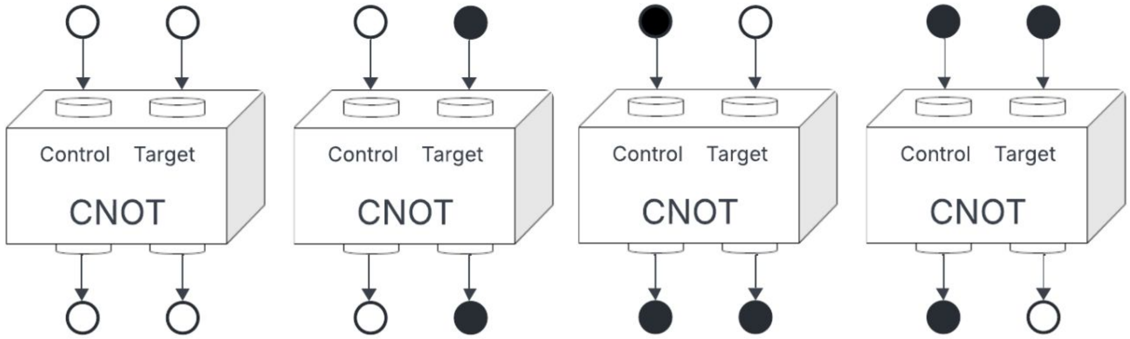


Figure 3. Behavior of the two qubit gate C-NOT.

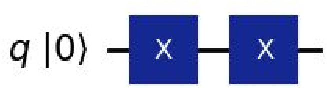
A Simple Circuit

Next, we consider that by stacking boxes on top of each other, we can use the output of one box as the input to another. For example, we can stack two NOT boxes, as shown on page 8 in [10] with the result that the output now matches the input: NOT(NOT(W)) = W and NOT(NOT(B)) = B. We assume familiarity with Python and Google Colab. So, naturally, our next step is to implement this arrangement in Qiskit (the X gate is the quantum NOT):

```
[1] !pip install qiskit pylatexenc
```

```
[2] from qiskit import QuantumCircuit
```

```
[3] qc = QuantumCircuit(1)
    qc.x(0)
    qc.x(0)
    display(qc.draw('mpl', initial_state=True))
```




```
[4] from qiskit.quantum_info import Statevector
    a = Statevector.from_instruction(qc)
    a.draw('latex')
```

$|0\rangle$

Figure 4. Stacking two NOT boxes in Python (Qiskit) recovers the input, as expected..

Access to the quantum emulator is immediate. We now introduce the Hadamard gate.

A Necessary Detour

The Hadamard gate is a fundamental single-qubit quantum gate used to create superposition states. In the book [10] it is known as the PETE box. As Terry indicates in the FAQ on the book's website [11] it is the only "actually quantum" box used in the book; all other boxes in the book just shuffle colors around and would be at home in a classical computer. This group of gates is universal; every calculation can be done (to good-enough accuracy, and perhaps with a small overhead) using only PETE boxes and the classical boxes. The reason is a remarkable mathematical result due to Shih, leveraging another powerful result due to Kitaev.“ There's a citation at the end of the book,” Terry says. “A few years ago I was in the middle of pondering this result when I realized I was running late to give a talk at a math camp for 12-14 year olds [...] run in part by my friend PETE Shadbolt. I raced for the tube, and while on it thought about what could I explain to these kids that wasn't the usual jargon-filled quantum fluff. And so here we are.”

Our goal in this paper is to introduce the misty state formalism from [10] in its pure form (no coefficients whatsoever) and use it to prove the phase kickback phenomenon. With the pure misty states formalism exactly as defined in the book one can show entanglement, Deutsch-Josza, Grover search, superdense coding, the GHZ game and entanglement swapping. It is true that one needs to extend this formalism to properly deal with phenomena such as W-entangled states (involving controlled-Hadamard gates and arbitrary rotations) and teleportation (since the input to the quantum teleportation algorithm is an arbitrary[3] quantum state) but at that point the extension feels natural. The misty formalism is universal, in as much as you can use it to do any quantum calculation with only a small overhead. “I should reiterate I am not advocating that we should recast all of quantum theory into this formalism. The misty state picture is a good way of getting people to the heart of some nontrivial quantum theory without them having to absorb a boatload of irrelevant math. But that math is not largely irrelevant if you actually want to work in the field, it makes many things much easier.” Math is our ultimate goal here as well. For example, we'd like our readers to be able and ready to read [1, 9, 13] as soon as they master the contents of our class.

We already have the NOT and C-NOT gates. We now define the Hadamard gate (also known as PETE box).

The PETE Box

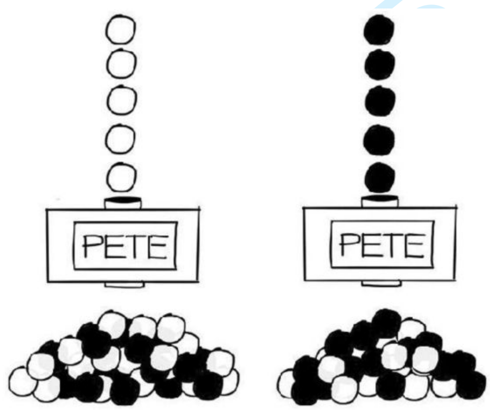


Figure 5. Behavior of the Hadamard gate (also known as the PETE box in [10]).

We capture the behavior shown in Fig. 5 by introducing the superposition operator. As a drawing it is represented as a cloud (hence, the name “misty state” used for a superposition state). This, again, would be a great opportunity to

[3] We emphasize then that we cannot clone but we can teleport an unknown, arbitrary quantum state.

watch (or rewatch) Terry's video [12] off the book's website at [11]. In text we can use the following two representations corresponding to each one of the situations shown above: H(W) = [W, B] and H(B) = [W, -B]. The notation says, in essence, that there are two outcomes and each one is equally likely to be measured. Here's how we represent these two transformations graphically:

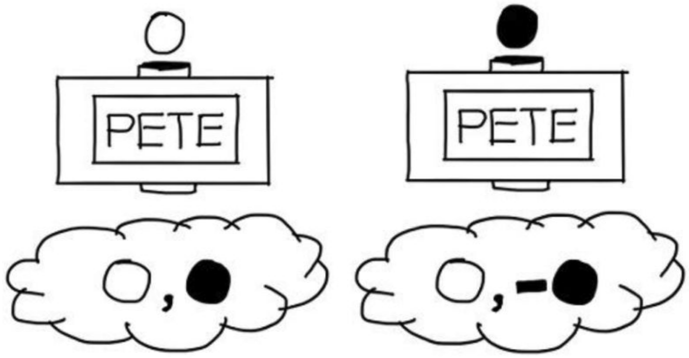


Figure 6. Misty states are superposition states (as shown in [10]). A negative sign (phase) shows on the right.

The Z Gate

The Z gate is introduced here as an exercise. Its definition is Z(W) = W and Z(B) = -B. Show that, just like for NOT, stacking two Z boxes leaves the input unchanged. We will soon learn that this is a general property of quantum gates and our next goal will be to prove it for PETE boxes (or Hadamard gates). That H(H(W)) = W and H(H(B)) = B is both non-trivial and very instructive. We can also demonstrate that experimentally in the Quantum Flytrap.

Exercise. Show that Z(H(W)) = H(X(W)) and Z(H(B)) = H(X(B)).

Linearity of Quantum Operators

A misty state, so far, is just a sum of two states with probability amplitudes equal to each other. The phase we encountered thus far is just a multiplication with the scalar -1. In quantum mechanics linearity of operators means that they satisfy two key properties: (a) they preserve the sum of states and (b) they preserve scalar multiplication. This property is fundamental to the superposition principle and how quantum states evolve over time. Therefore we shall enforce it here. As a result we have the following diagram showing how a NOT gate acts on a superposition of states:

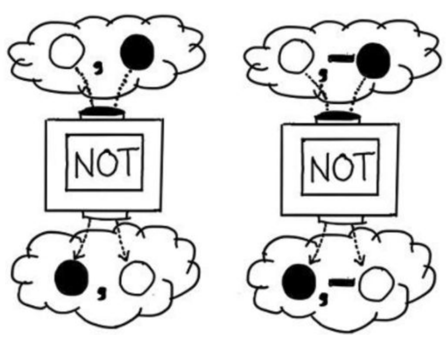


Figure 7. The effect of NOT gate on superposition of states. This diagram reproduced by permission from [10].

We can describe what happens in Figure 7 as follows:

NOT([W, B]) = [NOT(W), NOT(B)] = [B, W] = [W, B]

We take the opportunity to point out that like in a sum the order of factors (that is, of states in a superposition operator) does not matter so the NOT gate in effect leaves the first misty state unchanged.

In the case of the second diagram we have:

NOT([W, -B]) = [NOT(W), NOT(-B)] = [B, -NOT(B)] = [B, -W] = -[-B, W] = - [W, -B]

We have in fact proved that these are the two eigenvectors (CSCI: fixed points) of the NOT gate. In the process we illustrated linearity of phase and superposition operators with respect to the NOT gate. By a similar process we show how stacking two PETE boxes (or as everybody else knows them, Hadamard gates) leaves the input unchanged.

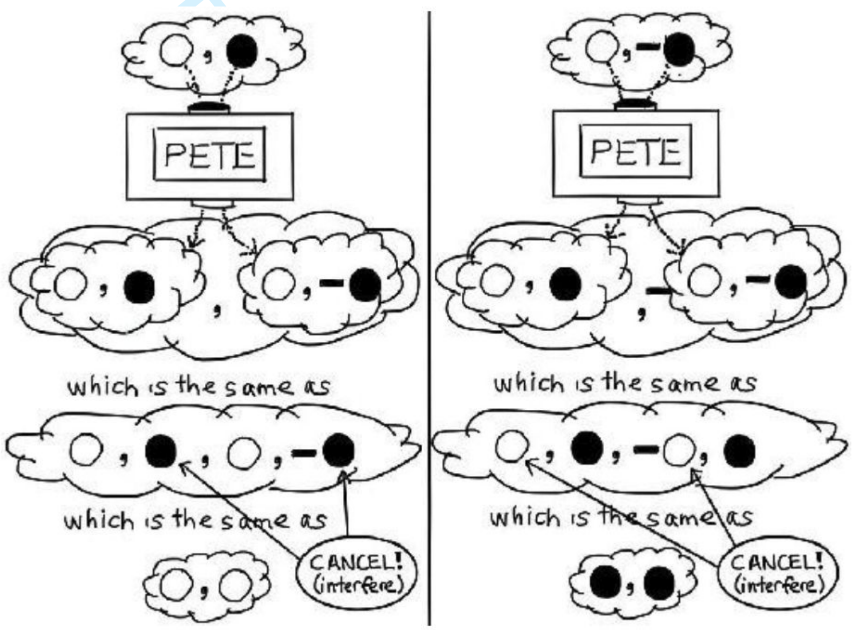


Figure 8. The effect of the PETE box (Hadamard gate) on two superpositions of states. Also from [10].

This part further uses the fact that a superposition operator is a sum and under certain conditions (i.e., when the superpositions are at the same depth and have the same number of distinct states) we can combine two mists by fading their boundaries so they can combine (join together) into a larger mist. Here's how this happens (in Fig. 8) in the notation we used to restate what was going on in Figure 7:

H(H(W)) = H([W, B]) = [H(W), H(B)] = [[W, B], [W, -B]] = [W, B, W, -B] = [W, W] = W

Now please look at Figure 8 as it shows (diagrammatically) what we wrote above, and below:

H(H(B)) = H([W, -B]) = [H(W), -H(B)] = [[W, B], [-W, B]] = [W, B, -W, B] = [B, B] = B

These relationships can be easily represented, reproduced and confirmed as experimental setups in Quantum Flytrap.

Systems of Two Qubits

There are four possible combinations of two qubits: WW, WB, BW and BB. We can represent this with white and black balls (or blobs) and we say that while they resemble multiplication they lack one important property of multiplication as they are not commutative. Thus WB and BW are different so order matters but other than that we can carry over some of the other properties encountered in multiplication: B [W, B] for example is the same as [BW, BB]. This, in effect, is how we define entanglement. Two (or more) particles are entangled when they are all described by the same wave function. For us this means that the expression that represents the state of the two (or more) qubits can't be separated as a product of factors each representing an individual qubit. Thus, because [BW, BB] = B [W, B] this equation does not describe a system of two entangled qubits. However a state like [BW, WB] cannot be split into a product of two states and thus represents an entangled state of two qubits (it's one of the Bell states). There is no entanglement in the Bernstein-Vazirani challenge that we discuss but we will be working with systems of two qubits so we wanted to clarify this up front.

The phase kickback is the following situation (that is, this is what we need to prove):

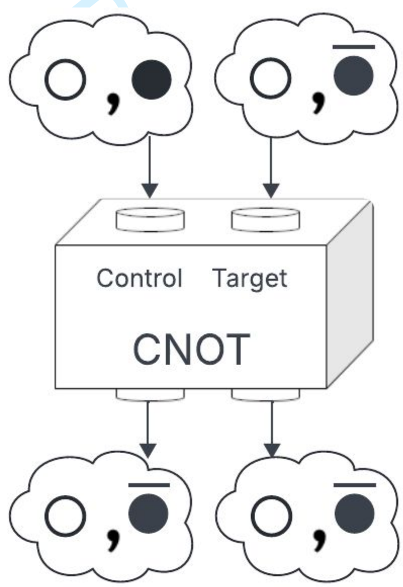


Figure 9. The phase of the target qubit is being "kicked back" to the first qubit. Target influences control.

How do we prove that that's what happening here? Let's start by writing the input as a system of two qubits. It is convenient here to keep the second qubit as a superposition and work with it as such. We have by distributivity:

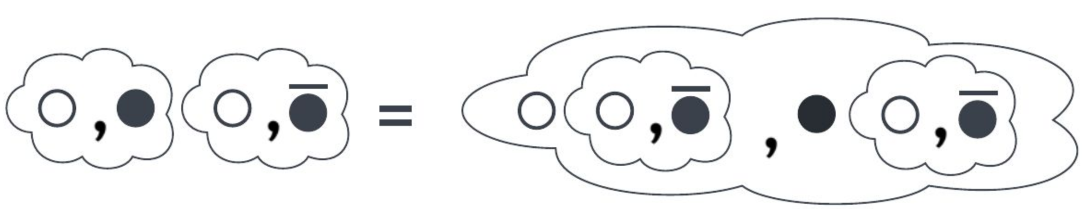


Figure 10. The input [W, B] [W, -B] = [W [W, -B] , B [W, -B]] in diagrammatic form.

Now that we have the input expressed as such let's pass it through the C-NOT gate and transform it using the rules of engagement already mentioned for this gate. We have:

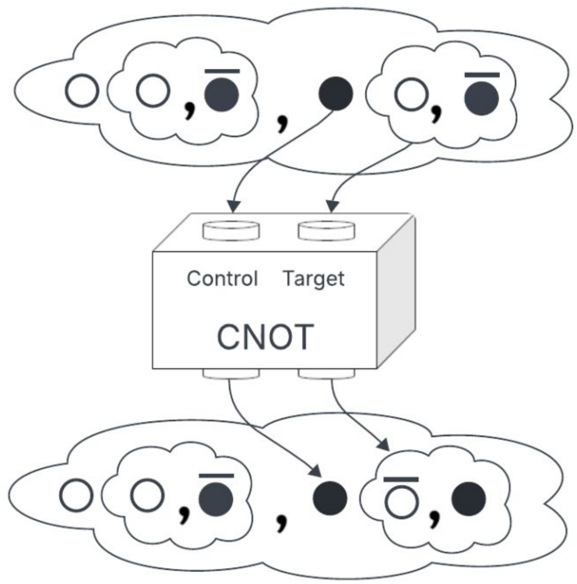


Figure 11. Effect of the C-NOT gate on our two qubit input.

As shown above, each pair of qubits passes through the C-NOT gate. The first one is placing a W on the control which means the gate will leave the second qubit unchanged. The second pair has a B on the control which flips the second qubit. The rule for flipping a superposition of states (via NOT) has been shown before and it's like in Figure 12, below.

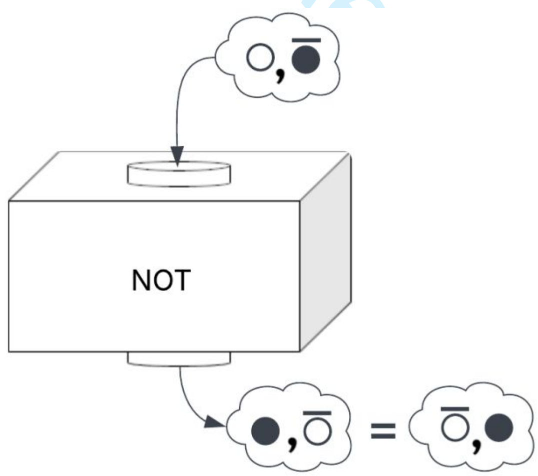


Figure 12. The effect of the NOT gate on a superposition of states. See also right side of Fig. 7.

The purpose of Fig. 12 is to support the transformation shown in Fig. 11. Thus, some readers might consider the picture to be be redundant while some might prefer to use shorthand to describe it, e.g. NOT([W, -B]) = [-W, B]. We've gone over this earlier when we said that the input here is one of the two eigenvectors of the NOT gate. Since the superposition operator is actually a sum (as we said before) the order of states in a mist is not important but an order is usually preferred and the phase distributes over the constituent states, as shown in the picture below:

Figure 13. A negative phase applied to a mist distributes over its constituent states.

Now we can rewrite the second state in the output of Figure 11 as follows:

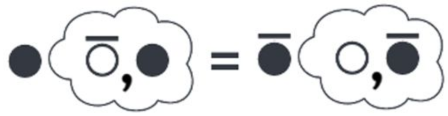


Figure 14. Moving the sign (phase) from the second qubit to the first has this effect.

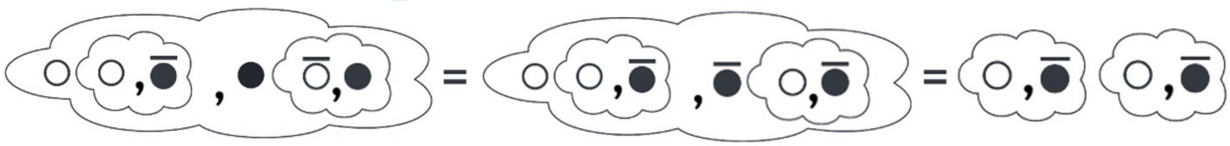


Figure 15. The output from C-NOT (factoring via reverse FOIL) as a product of states, each one a superposition.

So now we have proved the following:

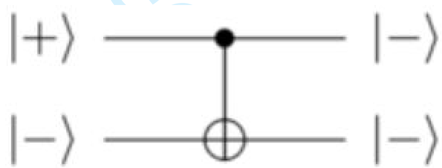


Figure 16. Phase kickback, conventional notation.

We’re now ready to solve the Bernstein-Vazirani challenge.

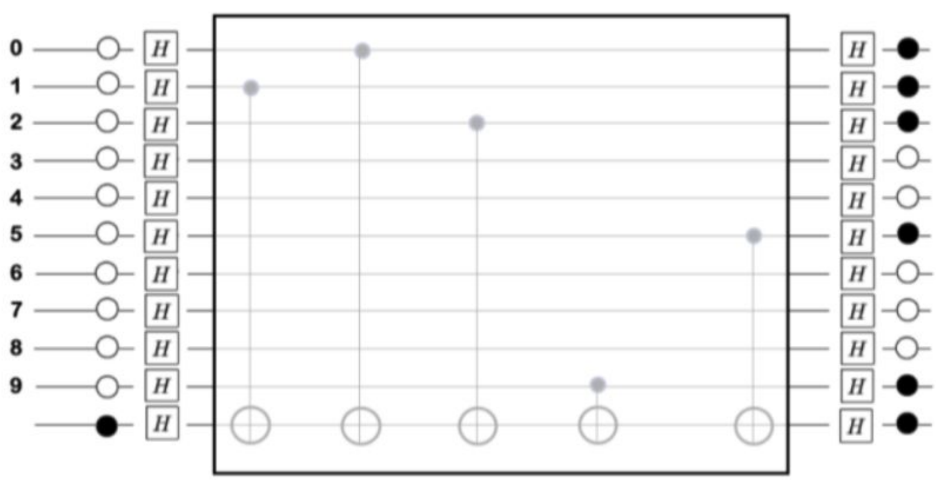


Figure 17. Bernstein-Vazirani challenge: the solution. If the C-NOT gates inside the black box had been oriented the other way the solution would have been immediate. But since they’re as shown above one would need to test every

input in part, thus establishing a linear lower bound for the complexity of finding the pattern. We trade space and hardware for speed. With the previous result and a corresponding number of Hadamard (PETE) boxes we can determine the structure of the black box in one step.

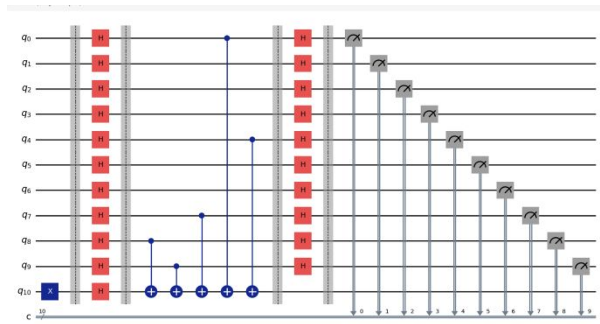
Below we show how this challenge can be implemented in Qiskit:

```
[1] !pip install qiskit qiskit-aer
```

```
[2] import qiskit
```

```
[3] from qiskit import *
    from qiskit_aer import AerSimulator
```

```
[4] secretnumber = '1110010001'
    indices = [1, 0, 2, 9, 5]
    n = len(secretnumber)
    circuit = QuantumCircuit(n+1, n)
    circuit.x(n)
    circuit.barrier()
    circuit.h(range(n))
    circuit.h(n)
    circuit.barrier()
    for index in indices:
      circuit.cx(n - index - 1, n)
    circuit.barrier()
    circuit.h(range(n))
    circuit.barrier()
    circuit.measure(range(n), range(n))
    circuit.draw(output='mpl')
```



```
[5] from qiskit_aer import AerSimulator
    device = AerSimulator()
    shots = 1
    job = device.run(circuit, shots = shots)
    result = job.result()
    counts = result.get_counts(circuit)
```

```
[6] print(counts)

{'1110010001': 1}
```

Figure 18. Creating and measuring the quantum circuit for the Bernstein-Vazirani challenge

From the previous figure we observe that the order of inputs is reversed in Qiskit so the circuit is reflected: as an example, the third C-NOT in the black box in our drawing connects the bottom line (its target) to the third line from the top (the control). In the Qiskit circuit it connects the bottom line (target) with the third line from the bottom (Qiskit numbers the qubit lines in reverse order). Also, the left-to-right order of gates in the black box is not important; but the vertical order of the lines in the input and output is—and the output is determined, as predicted, in one shot.

Conclusions and Acknowledgments

CS2023 makes some excellent recommendations on (for the first time ever) how to include a knowledge unit on quantum information science, computing and quantum algorithms. Their proposal is organized in three stages and comprises a short (eight-weeks) class, a one semester class and a longer, two semester sequence that (at least in principle) makes heavy use of a lab (or fab, depending on resources) in quantum hardware, gates and circuits. Following those recommendations we have described here our approach of implementing the eight-week syllabus with extended material from Terry Rudolph's groundbreaking "Q is for Quantum". This material has been tested in the classroom, in various conferences in workshops and tutorials, and at many levels – including creating a faculty learning community (FLC) for HS and middle school CSCI teachers in the state last summer with significant support from the Computer Science Teachers' Association (CSTA) in our state.

Afterword

We started with frisbees and classical mechanics and extolled the virtues of experiential learning. We then discussed what absolutely small means (according to Dirac) and how direct interaction with the quantum world does not come easy. Turing machines and lambda calculus were invented by Alonzo Church and his student Alan Turing; they're the cornerstones of our field and were introduced in the first part of the 1930s so they're younger than quantum mechanics. Quantum computing is the only model of computation that breaks the strong (extended) Church-Turing thesis. The misty state formalism favors learner-sighted teaching techniques and the Quantum Flytrap along with Python coding with Qiskit in Google Colab notebooks supports maker-centered learning. In this paper we solve a Bernstein-Vazirani challenge using the pure misty state formalism introduced in [10]. We only need the NOT gate, the two-qubit C-NOT gate, the mysterious PETE box (also known as the Hadamard gate) and the concept of superposition. Systems of two or more qubits can exhibit entanglement but this phenomenon is not present in the Bernstein-Vazirani challenge. The Z gate is introduced as a means to practice with the new system. Linearity of quantum operators translates into very simple operations with B and W (black and white balls). The system devised by Terry Rudolph is incredibly effective and works even with states that break the representation chosen in the book. We would have loved to show you irreducible misty states (e.g., the two Hadamard eigenvectors have this property) and how useful they are in facilitating abstraction and representation of non-classicality. More exercises would have asked you to design a one-qubit circuit that generates the eigenvectors of the Hadamad gate and then a two-qubit circuit that would do the same. Unfortunately space restrictions did not allow us to share those with you here. A longer version of this paper is available online [5].

References

1. Burd, B. *Quantum Computing Algorithms: Discover how a little math goes a long way.* Packt Publishing, Sep 22, 2023. First edition.

2. Ekert, A. (adviser) Quantum Flytrap: A No-Code IDE for Quantum Computing. https://quantumflytrap.com/ Accessed: 2025 May 21.

3. Ellwanger, D. *A Quantum Computing Pamphlet*. Independently published, Sep 8, 2019.

4. Fayer, M.D. *Absolutely Small: How Quantum Theory Explains Our Everyday World.* Amacom Books, Special Edition, June 16, 2020.

5.

6. Gigerenzer G. Embodied Heuristics. Frontiers in Psychology 2021 Sep 30 https://tinyurl.com/embodied-heuristics URL accessed successfully on 2025 May 21

7. Haldane, A.G. and Madouros, V. The Dog and the Frisbee. Presented at the Fed. Reserve Bank of Kansas City's annual economic policy symposium Aug 30-Sep 1, 2012 in Jackson Hole, Wyoming. https://tinyurl.com/frb-of-kc, has link to the talk/presentation/materials: https://tinyurl.com/the-dog-and-the-frisbee both accessed: 2025 May 21.

8. Marinescu, D.C. and Marinescu, G.M. *Approaching Quantum Computing*. Pearson Education Inc., Sep 20, 2004.

9. Nielsen, M.A. and Chuang, I.L. Quantum Computation and Quantum Information. Cambridge University Press, January 31, 2011. Tenth anniversary edition.

10. Rudolph, T. *Q is for Quantum*. Published by Terence Rudolph, July 19, 2017.

11. Rudolph, T. Book website: https://www.qisforquantum.org accessed: 2025 May 21.

12. Rudolph, T. https://tinyurl.com/terry-rudolph-misty-states accessed: 2025 May 21

13. Watrous, J. Understanding Quantum Information and Computation (IBM Quantum Education Series of Video Lectures) https://tinyurl.com/watrous successfully accessed on 2025 May 21.

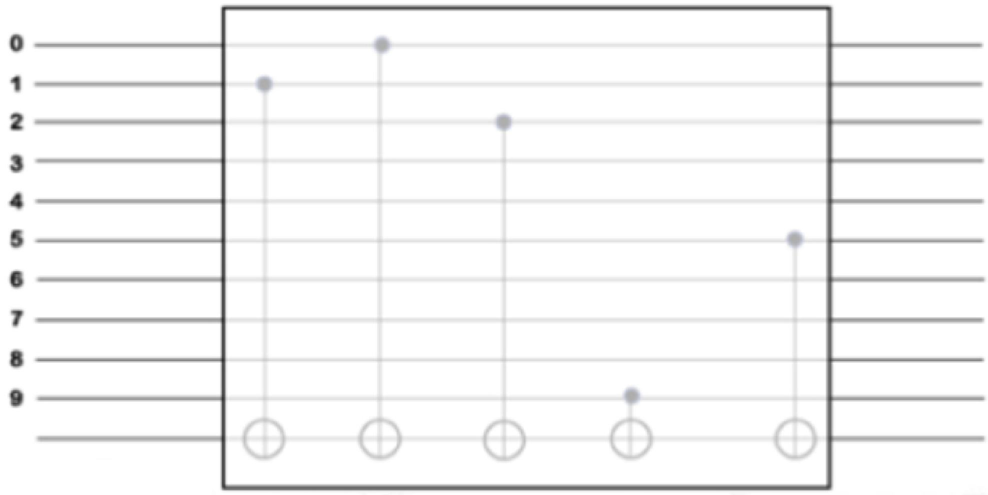


Figure 1. A Bernstein-Vazirani challenge. The black box contains C-NOT gates; they are introduced below.

243x121mm (72 x 72 DPI)

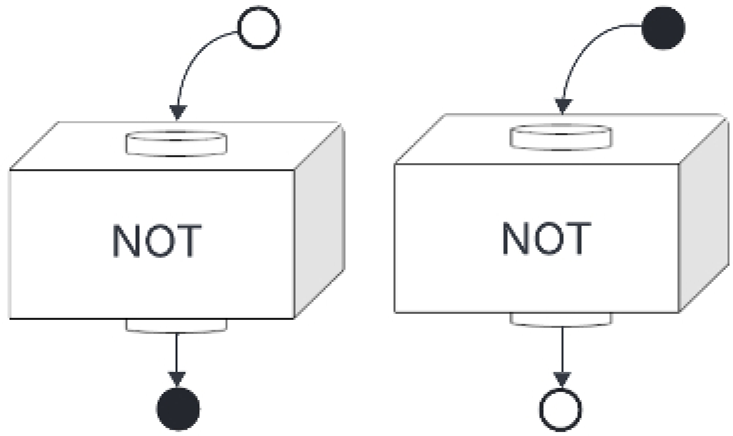


Figure 2. The NOT gate flips its input: NOT(W) = B and NOT(B) = W.

131x79mm (72 x 72 DPI)

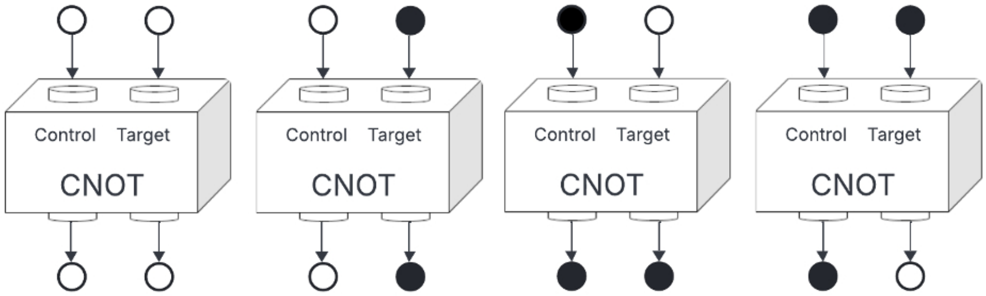


Figure 3. Behavior of the two qubit gate C-NOT.

330x106mm (72 x 72 DPI)

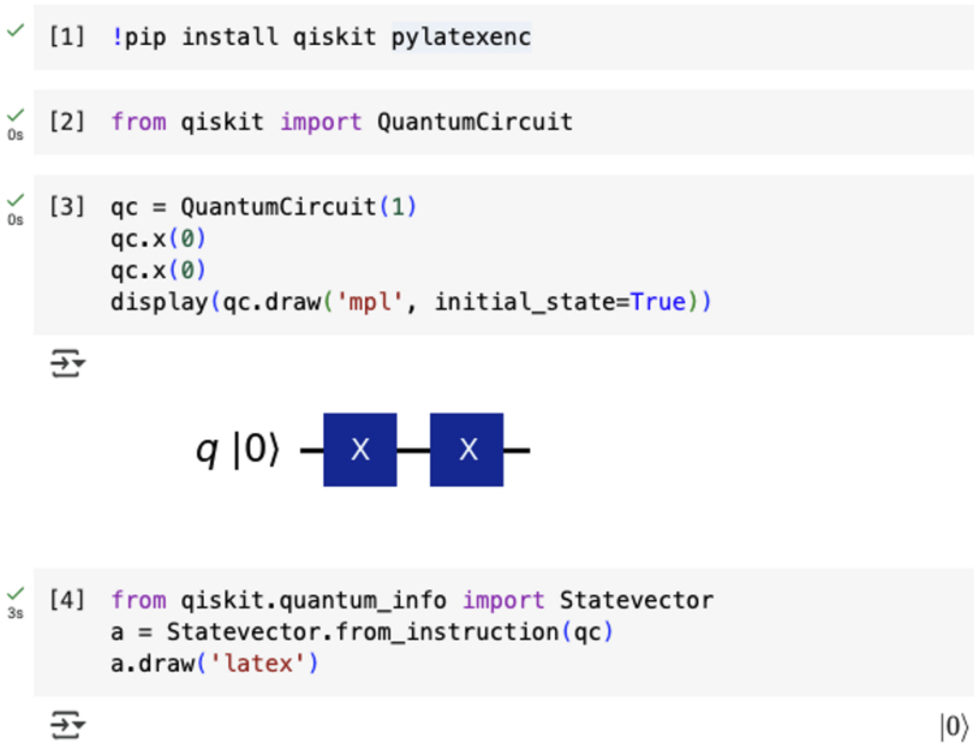


Figure 4. Stacking two NOT boxes in Python (Qiskit) recovers the input, as expected.

270x207mm (72 x 72 DPI)

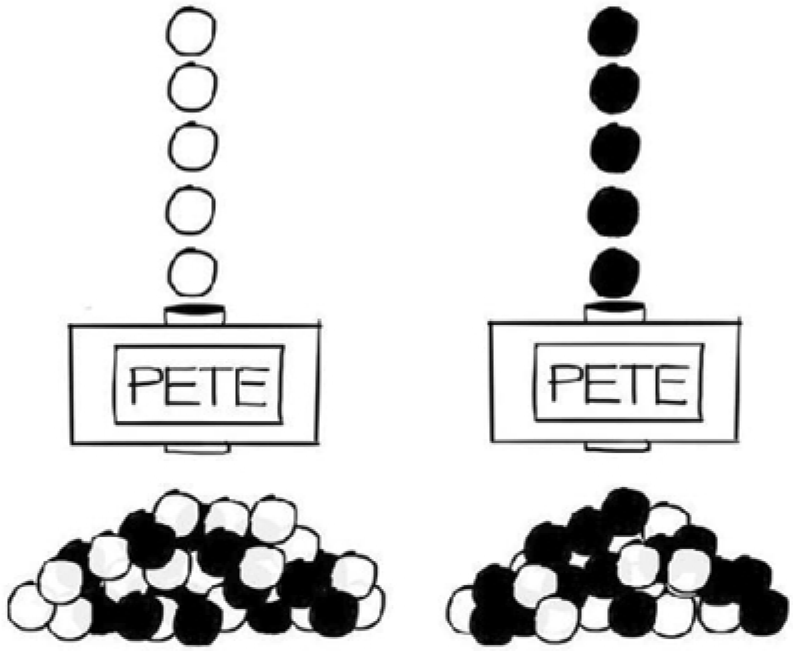


Figure 5. Behavior of the Hadamard gate (also known as the PETE box in [10]).

154x121mm (72 x 72 DPI)

Figure 6. Misty states are superposition states (as shown in [10]). A negative sign (phase) shows on the right.

201x106mm (72 x 72 DPI)

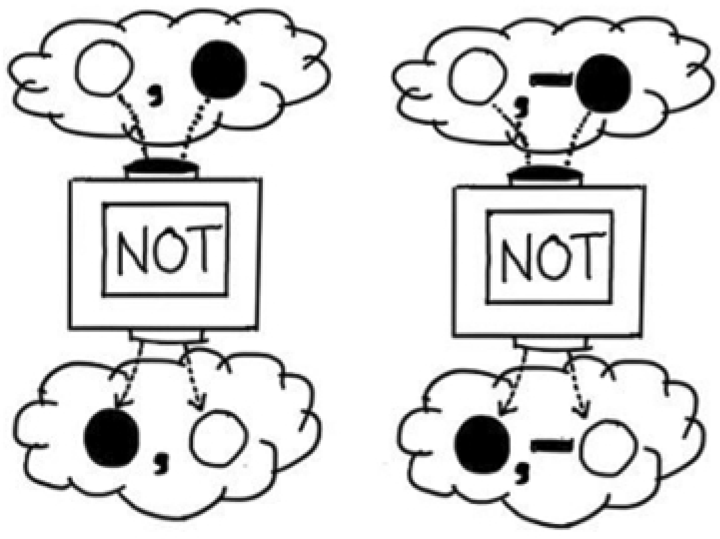


Figure 7. The effect of NOT gate on superposition of states. This diagram reproduced by permission from [10].

130x99mm (72 x 72 DPI)

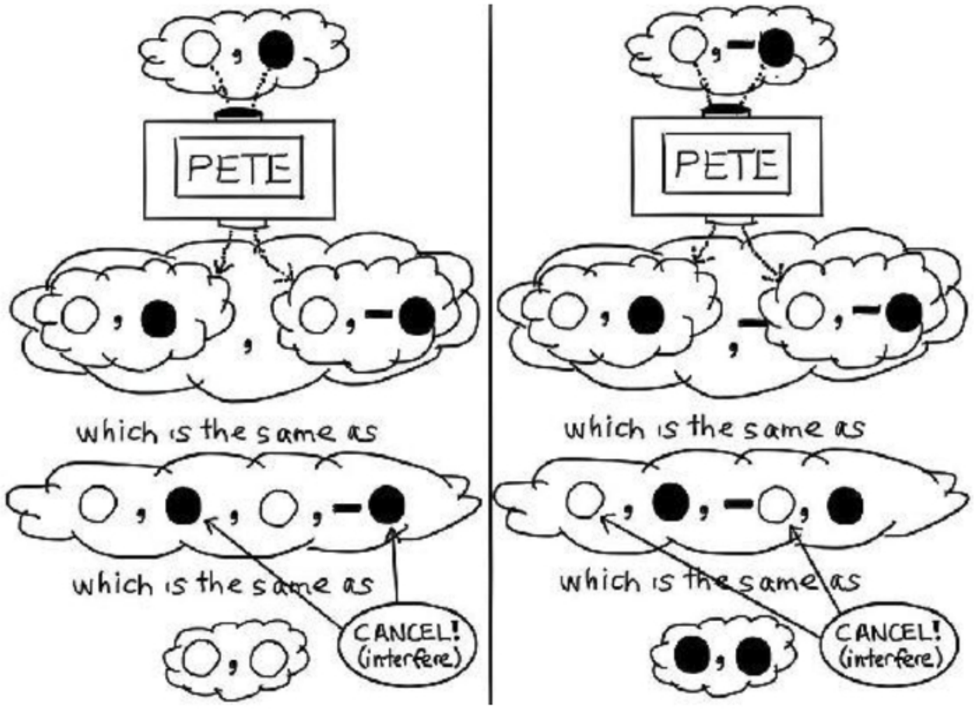


Figure 8. The effect of the PETE box (Hadamard gate) on two superpositions of states. Also from [10].

252x184mm (72 x 72 DPI)

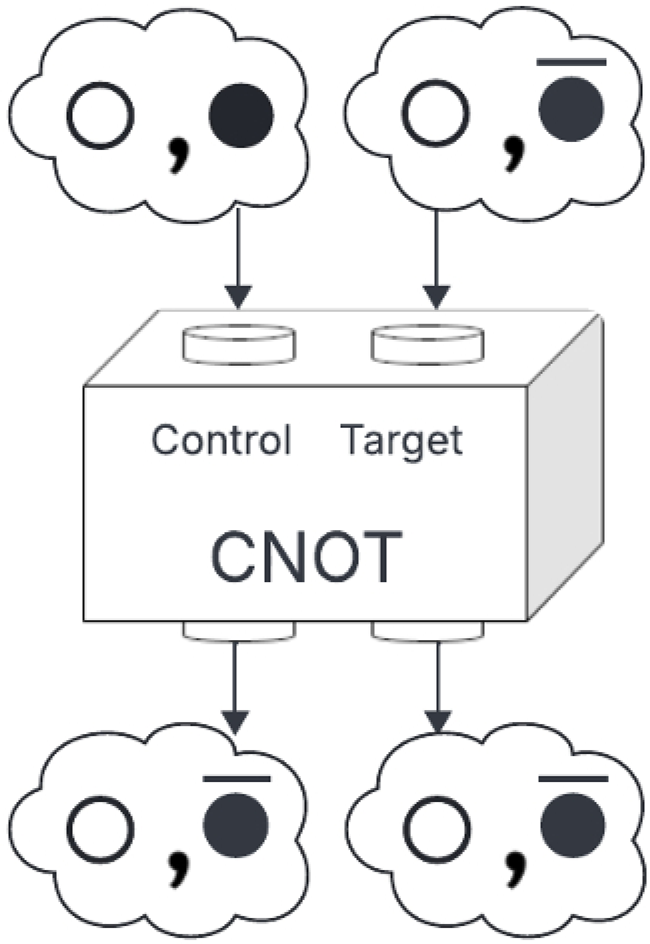


Figure 9. The phase of the target qubit is being “kicked back” to the first qubit. Target influences control.

125x170mm (72 x 72 DPI)

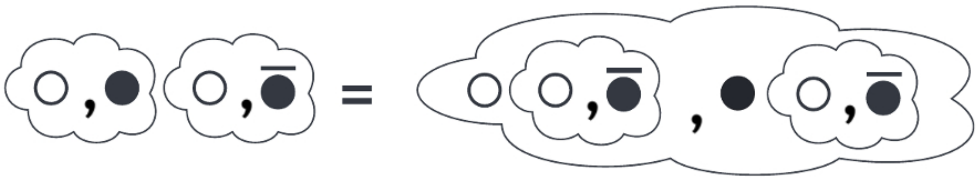


Figure 10. The input [W, B] [W, -B] = [W [W, -B] , B [W, -B]] in diagrammatic form.

318x63mm (72 x 72 DPI)

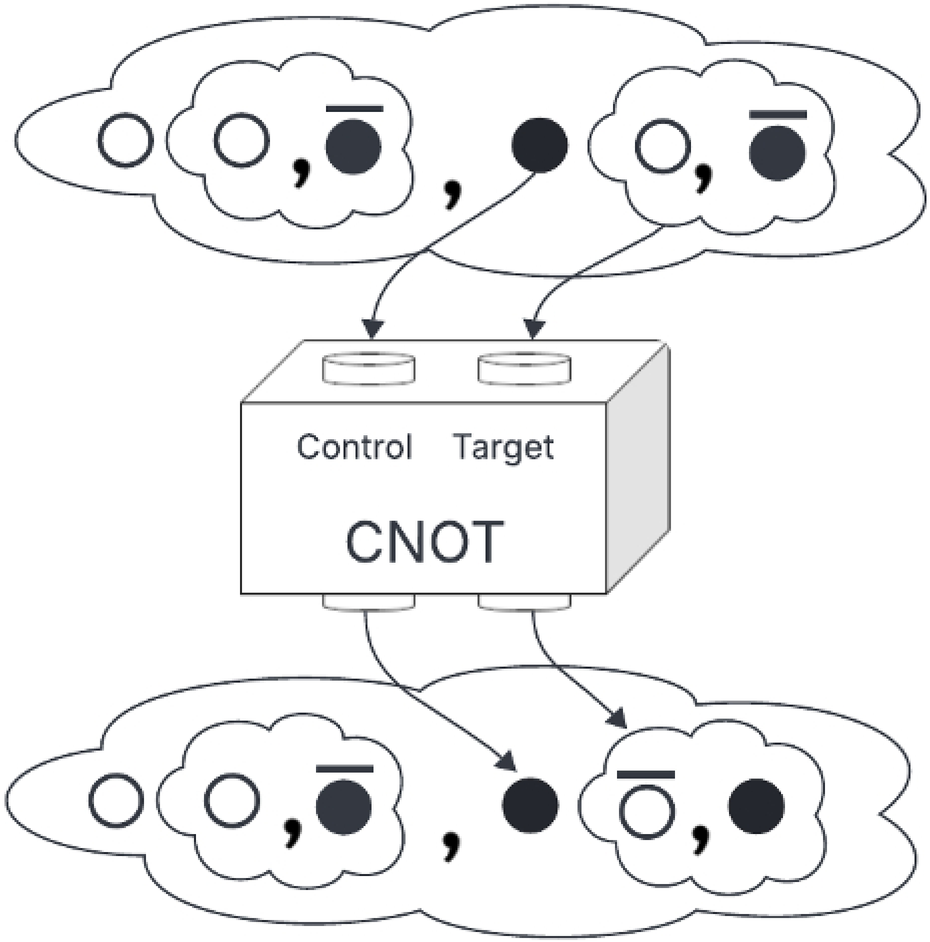


Figure 11. Effect of the C-NOT gate on our two qubit input.

176x169mm (72 x 72 DPI)

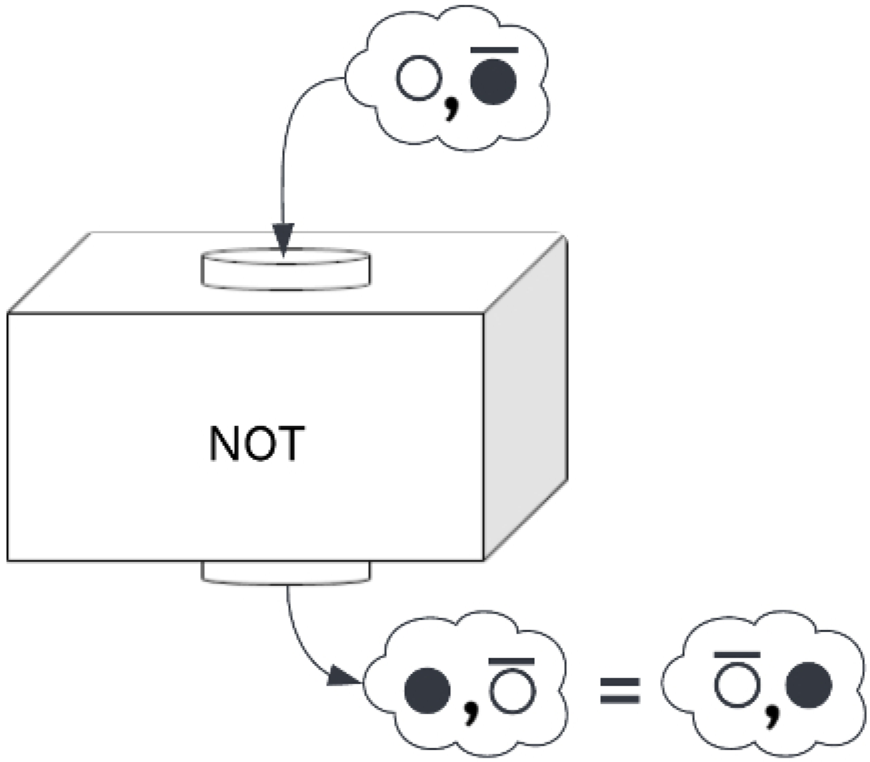


Figure 12. The effect of the NOT gate on a superposition of states. See also right side of Fig. 7.

158x137mm (72 x 72 DPI)

Figure 13. A negative phase applied to a mist distributes over its constituent states.

178x40mm (72 x 72 DPI)

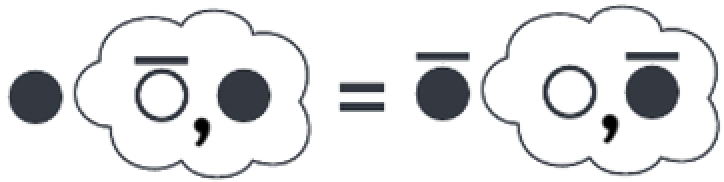


Figure 14. Moving the sign (phase) from the second qubit to the first has this effect.

134x34mm (72 x 72 DPI)

Figure 15. The output from C-NOT (factoring via reverse FOIL) as a product of states, each one a superposition.

359x45mm (72 x 72 DPI)

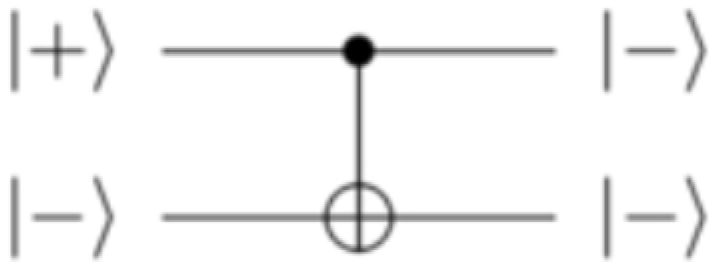


Figure 16. Phase kickback, conventional notation.

135x56mm (72 x 72 DPI)

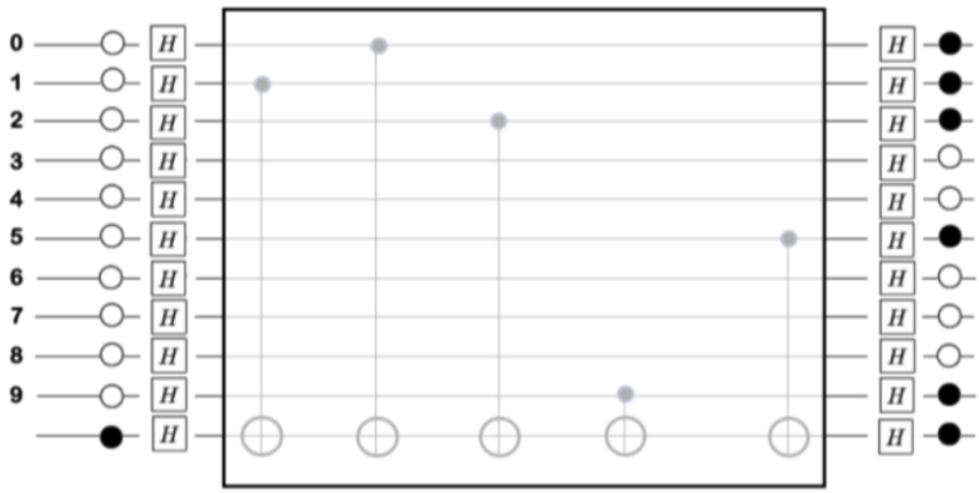


Figure 17. Bernstein-Vazirani challenge: the solution. If the C-NOT gates inside the black box had been oriented the other way the solution would have been immediate. But since they're as shown above one would need to test every input in part, thus establishing a linear lower bound for the complexity of finding the pattern. We trade space and hardware for speed. With the previous result and a corresponding number of Hadamard (PETE) boxes we can determine the structure of the black box in one step.

275x143mm (72 x 72 DPI)

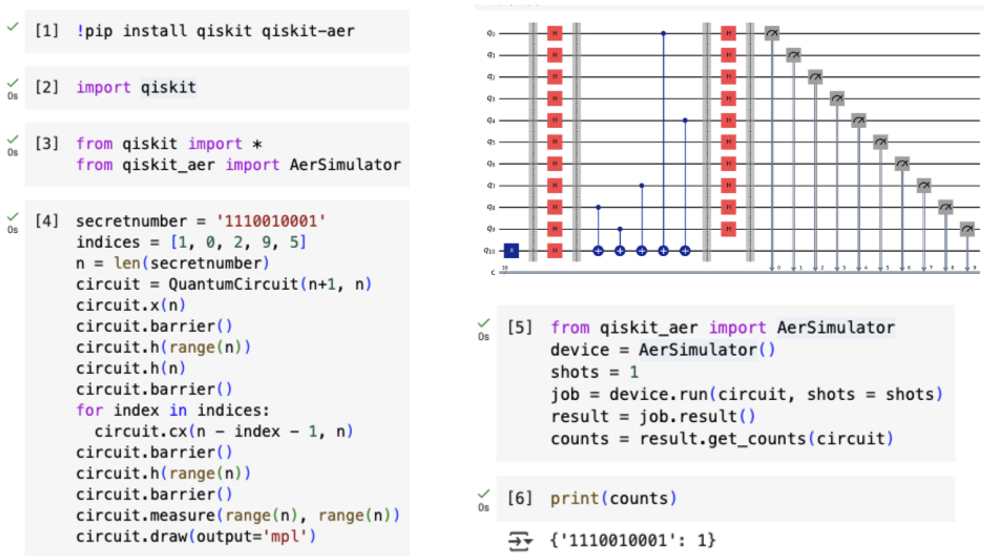


Figure 18. Creating and measuring the quantum circuit for the Bernstein-Vazirani challenge

335x192mm (72 x 72 DPI)