On the Design and Implementation of a Quantum Architectures Knowledge Unit for a CS Curriculum

Adrian German
dgerman@indiana.edu
Indiana University
Bloomington, Indiana, USA

Marcelo Pias
Federal University of Rio Grande
Rio Grande, Brazil
mpias@furg.br

Qiao Xiang
Xiamen University
Xiamen, Fujian, China
xiangq27@gmail.com

Figure 1: Schematic overview of the three curricular plans with specific sample instantiations.

ABSTRACT

Sixteen years ago, Scott Aaronson remarked (in the presence of Ray Laflamme) that quantum mechanics (QM) resembles an operating system on which the rest of Physics is running its application software (except for general relativity "which has not yet been successfully ported to this particular OS"). Prior to that, it took the insight of an educator and eminent computer scientist (Umesh Vazirani) to realize that a complete and consistent introduction to QM can be given via the language of qubits and quantum gates. Closer to the present, it took the profound intuition of another polymath (Terry Rudolph) to realize that the linear algebra normally at the foundation of such an approach can be replaced with a simple rewriting system accessible to middle school students. Rewriting systems are at the foundation of Computer Science, they are, in fact, the very fabric of it (e.g., Turing machines and lambda calculus), so these are very fortunate developments. Furthermore, a linear algebra prerequisite is now shared firmly in the CS undergraduate curriculum with Machine Learning, a topic that has known a very deep and sudden revival. Quantum Information Science and Technology (QIST) is inherently interdisciplinary and spans physics, computer science, mathematics, engineering, chemistry and materials science. We present three curricular plans for incorporating QIST topics (via Quantum Computing) into the CS undergraduate curriculum. Such plans have been constructed with a preliminary consultation with QED-C members (industry, academia, national labs, and government agencies) asking for comments, suggestions and general input on these three curricular plans.

CCS CONCEPTS

- **Hardware → Quantum computation**.

KEYWORDS

quantum information science, quantum computation, quantum processing unit, undergraduate curriculum, quantum architectures

ACM Reference Format:
Adrian German, Marcelo Pias, and Qiao Xiang. 2023. On the Design and Implementation of a Quantum Architectures Knowledge Unit for a CS Curriculum. In *Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 1 (SIGCSE 2023), March 15–18, 2023, Toronto, ON, Canada.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3545945.3569845

1 INTRODUCTION

From the outset, we need to distinguish between Quantum Mechanics (QM) and Quantum Information Science and Technology (QIST). QM is the science describing the behavior of matter and light on the atomic and subatomic scale. QIST is an emerging interdisciplinary academic discipline concerned with studying the new possibilities QM offers for acquiring, transmitting, and processing information. Fields under QIST include quantum computing (QC), quantum sensing (ultrasensitive precision measurements), quantum communication, quantum cryptography, materials for quantum information and more. Specifically, quantum computing is a type of computation whose operations can harness quantum mechanical phenomena such as *superposition, interference and entanglement.*

SIGCSE 2023, March 15–18, 2023, Toronto, ON, Canada

ACM ISBN 978-1-4503-9431-4/23/03…$15.00
https://doi.org/10.1145/3545945.3569845

In early 2020, the National Science Foundation invited 25 researchers and educators to come together to deliberate on defining a core set of key concepts for future QIS learners. The "Key Concepts for Future QIS Learners" workshop [35] was initially scheduled to happen at Harvard University in March 2020, hosted by the NSF Center for Integrated Quantum Materials (CIQM). Reconceived as a virtual workshop, it took place over three weeks, followed by several weeks of reporting, editing, reviewing and consolidation by a core team of workshop participants. The workshop report [15] identified a core set of nine key concepts for future QIS learners that could provide a starting point for further curricular and educator development activities.

Worldwide, there is growing excitement, investment, and competition in the area of QIST. A technological ecosystem is being shaped by public and private investment in North America, Asia and Europe. The potential implications of QIST are broad; quantum technology may eventually underlie a new technological infrastructure, much like the semiconductor revolution changed everything in the second half of the last century. With this growth, there has been a steady demand for QIST-trained professionals. Recognizing that the impact on the CS undergraduate curriculum is imminent, our paper aims to answer the following questions:

1. What is the minimum set of topics (knowledge unit - KU) that a CS undergraduate should be taught in a Quantum Architectures course?
2. What is the interface for this knowledge unit (where input: required skills, and output: learning outcomes)?
3. What is the tension between disciplines such as math, physics, engineering and computer science? How do we get a genuinely trans-disciplinary learning program off the ground?
4. What is the theoretical minimum that captures the essence without overwhelming details in physics and maths?
5. How can we ensure a 10-15 years horizon shelf life for this knowledge unit (including concrete recommendations for a CS undergraduate "quantum hardware lab")?

The rest of the paper is structured as follows: we start by discussing the extent to which a CS undergraduate needs to be exposed to QM concepts. We compare and contrast with related, recent proposals for Undergraduate Quantum Engineering Programs and Degrees [2, 12]. We then enumerate options available at the high school and even middle-school level. These could form the basis for an eight-week (half-semester) undergraduate class tailored to CS students. Existing workshops for high-school teachers and industry certificate programs for workforce development, training or upskilling are reviewed, and relevant literature is presented in the larger context. These could help inform curriculum development for community colleges within a national quantum technician consortium. We then discuss our main proposal: a lab-based full-semester computer-assisted class on the most enduring quantum computing concepts. In the process, we review the qubit modalities and enumerate the characteristics of a Quantum Processing Unit (QPU). Implementation aspects are being discussed along with the need to address error mitigation and (classical) control—topics relevant to the current noisy, intermediate-scale quantum (NISQ) devices of our time—before we conclude with a summary section.

2 QUANTUM MECHANICS (QM)

QIST has three pillars: quantum sensing, quantum networking and quantum computing (collectively known as quantum technology).

The learner should realize that we can address quantum technology only because of QM. And the reason for QM is Nature. Thus, from the start, the learner needs to develop a qualitative appreciation for the quantum nature of our world. Understanding what we see around us [48] requires some knowledge of quantum mechanics: color is a quantum phenomenon; energy levels and color come from the wavelike nature of particles; quantum phenomena hold atoms together to make molecules and determine their shapes; carbon dioxide is a greenhouse gas because of quantum effects; very hot objects give off visible black body radiation; electrical heating is a quantum phenomenon, and so on.

As early as the 1910s, two different physicists[1] had proved that a permanent magnet cannot exist in classical mechanics at a finite temperature. So every time someone uses a magnet to put a note on their refrigerator, they use quantum mechanics. Quantum engineers – and to some extent the new generation of computer scientists (literate, articulate or ultimately fluent in their knowledge) – should be able to answer questions such as: Why can we see through glass but not steel? If all matter is mostly empty space, what prevents me from putting my hand through the wall? What is so special about the diffraction used by DVDs, or the partial reflection of light, or about magnets? Why do metals conduct electricity when glass does not[2]? Why does a piece of metal glow red when it is hot and turn blue when it is even hotter? How do smoke detectors work[3]? What makes salt dissolve in water while oil does not? Why are strawberries red and blueberries blue? How do the transistors in modern computer chips act as switches[4]? How do atomic clocks work? How about superconductors and superconducting magnets? How does the MRI machine work[5]?

Quantum mechanics is an abstract theory that explains all of reality. This is a hard sell to a computer scientist or an engineer who has just spent four years not learning about it. But, if reality is quantum, then what is the language of quantum? And our question as quantum computer scientists or engineers is: can we make artificial systems that behave more like nature (quantum mechanical)? Can we make artificial systems like the ones already existing in nature, that operate on the principles of quantum mechanics? Can we control and program such artificial systems and make them do what we want? If so, what would be the benefit of such an exercise? It would be related to solving problems in physical sciences that have prohibitive complexity for the current classical approaches.

2.1 A Quantum Intuition

Dirac taught us that a minimum disturbance accompanies a measurement (inherent in the nature of things, and that cannot be overcome by improved experimental technique). If the minimum

---

[1] Niels Bohr (1911) and (independently, in 1919) Hendrika Johanna van Leeuwen.

[2] Answer: a sea of electrons created by Pauli exclusion principle.

[3] Hint: alpha particles create a constant current that changes when smoke enters the chamber.

[4] Answer: they use quantum tunnelling.

[5] Even qualitative, non-mathematical answers to some of these questions would prove quintessential to the education of a CS undergraduate involved in building any quantum computing hardware in the lab.

disturbance accompanying a measurement is non-negligible, then the object is absolutely small, and its properties fall in the realm of quantum mechanics. The quantum properties of tiny particles are not strange; they are just unfamiliar and not subject to our classical intuition. Can any intuition be developed at all?

Styer's books [46, 47] provide a clear, accessible introduction to the topic. Other books [17, 39, 51] explain QM to incoming college students using only a minimum of pre-calculus or do not resort to any mathematics at all. But even books heavy on maths point out [33] that "nonphysicists have a particular problem finding a suitable introduction to the subject". In recent years, the Massive Open Online Courses (MOOC) allowed creative instructors to develop innovative programs, and many are available online as we write this—for example, on EdX [20, 21]. An excellent (calculus-based) introduction to QM that caters to the needs and motivations of the CS undergraduate student can be found here [8].

We argue that students need to be exposed to quantum mechanics. The actual extent (on a spectrum ranging from [34] to [10]) should be the learner's decision, possibly with the guidance of an academic adviser. The curriculum should provide adequate options that could be instantiated in course-based activities.

It is worth observing that with quantum technology, engineers are starting to build things from the bottom up, atom by atom, electron by electron. This represents a fundamental shift [4, 5] in how humanity builds things. This type of engineering, in the realm of tiny particles, deserves a new kind of engineering intuition that can only be created by studying quantum mechanics.

2.2 A Computer Science Perspective

Can an adequate intuition about QM be developed from a mathematical standpoint? Absolutely. Mathematicians do that all the time. Mathematics is outstanding at enumerating and summarizing facts in very concise ways. Mathematical theorems are just ways in which one builds higher-level languages and concepts of increased expressive power. A quote from Scott Aaronson [1, 23] adds an interesting perspective to this discussion:

> "I like to say that, after all the forbidding–sounding verbiage you read in popular books, quantum mechanics is astonishingly simple—once you take the physics out of it! In fact, QM isn't even 'physics' in the usual sense [... QM i]s a certain generalization of the laws of probability. It says nothing directly about electrons, photons, or anything like that. It just talks about lists of complex numbers called amplitudes: how these amplitudes change as a physical system evolves, and how to convert them into the probability of seeing this or that result when you measure the system."

It may seem a bit arrogant, but such pragmatic points of view have always been a staple of our field:

> "And everything you've ever heard about the 'weirdness of the quantum world,' is simply different logical consequences of this one change to the rules of probability. This makes QM, as a subject, possibly more computer-science friendly than any other part of physics. [...] In fact, even if our universe hadn't been described by QM, I suspect theoretical computer scientists would've eventually needed to invent quantum computing anyway, just for internal mathematical reasons. Of course, the fact that our universe is [in fact, quantum mechanical] does heighten the interest!"

3 QUANTUM COMPUTING

Classically, the time it takes to do specific computations can be decreased by using parallel processors. To achieve an exponential decrease in time requires an exponential increase in the number of processors and hence an exponential increase in the amount of physical space needed. In quantum systems, the amount of parallelism increases exponentially with the size of the system. Thus, an exponential increase in parallelism requires only a linear increase in the amount of physical space needed. This effect is called quantum parallelism [40, 41]. However, there is a catch: while a quantum system can perform massive parallel computation, access to the computation results is restricted. Accessing the results is equivalent to making a measurement, which disturbs the quantum state. This problem makes the situation seem even worse than the classical situation; we can only read the result of one parallel thread, and because the measurement is probabilistic, we cannot even choose which one we get. But in the past few years, a few researchers have found efficient ways of finessing the measurement problem to exploit the power of quantum parallelism. This type of manipulation has no classical analog, and requires non-traditional programming techniques.

In the last twenty years, several meaningful attempts have been made (in print form: papers and books) to guide computer scientists through the barriers that separate quantum computing from conventional computing. A sequence of steps that have proved to be very successful starts from the basic principles of QM to explain where the power of quantum computers comes from and why it is difficult to harness. One then describes quantum cryptography, teleportation, and dense coding. Various approaches to exploiting the power of quantum parallelism are also explored before one concludes with a discussion of quantum error correction. We now examine various potential paths from the outskirts to the center of this approach.

3.1 A High-School Physics Perspective

The reference here is [24], a book also available for free download through Springer's Open Access [25]. This classroom-tested textbook uses simple language, minimal math, and plenty of examples to explain the three key principles behind quantum computers: *superposition, quantum measurement, and entanglement.* It then describes how this quantum world opens up a new computing paradigm. The book bridges the gap between popular science articles and advanced textbooks by making key ideas accessible with just high school physics as a prerequisite. In addition, each unit is broken down into sections labelled by difficulty level, allowing the course to be tailored to the student's experience of math and abstract reasoning. Problem sets and simulation-based labs of various levels reinforce the concepts described in the text and give the reader hands-on experience running quantum programs.

At the high school level, the book can be used after the AP or IB exams, in an extracurricular club, or as an independent project resource to give students an introduction to quantum computing. At the college level, it can be used as a supplementary text to enhance a variety of courses in science and computing or as a self-study guide for students who want to get ahead. Additionally, business, finance, or industry readers will find it a quick and helpful primer on the science behind computing's future.

3.2 A High-School Precalculus Perspective

The reference here is [6], and the text is again available for free from [7]. This book introduces quantum computing using only high school algebra and trigonometry. It presents topics from quantum mechanics, algebra, computer science and cryptography, which form the foundation of the quantum computing theory.

Understanding Shor's algorithm is one of the goals of this book. Note that the book avoids entirely complex numbers but it is otherwise mathematically very thorough, sound and clear.

This book and the one presented in the previous section would make great texts for early college graduates' eight-week (half-semester) classes. We now focus on a third book that could be used to start a longer (full-semester) course of study that would turn to linear algebra around the midterm.

3.3 Quantum Abacus with Misty States

Remarkably, Terry Rudolph [42] is able to introduce most of quantum mechanics using only elementary algebra. Many of the paradoxes and the problems of interpretation of quantum mechanics can be discussed at this level, and the author does this using a very personal point of view. In Terry's [43] own words:

> 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"). [...] It is possible to replace linear algebra with some string-rewriting rules which are no more complicated than the basic rules of arithmetic.

What is remarkable about this is that this approach has already spawned several successful and creative projects. One example is the [3, 14] very popular High-School-level Summer School for Quantum Information Science and Engineering at Virginia Tech. Other examples include the two MOOC courses designed by Diana Franklin [18, 19] from the University of Chicago available in EdX as we write this. Anyone can enroll and take these classes and, going through videos, text and plenty of challenging exercises, and if they score above 70% obtain a certificate. What is remarkable about these two MOOCs is that they not only do not cut any corners, but they make a very smooth transition from the Terry Rudolph's approach to the more traditional way (e.g., Vazirani) that is based on linear algebra.

4 QUANTUM INFORMATION SCIENCE

Quantum sensing has been with us for over 50 years and with incredible, truly remarkable success. Meanwhile, quantum computing (as a theoretical construct) has been around for only a little over than 30 years but in the last 15 years significant advances in quantum technologies have brought about a new, acute awareness about quantum computing. So we provide some references for some of the areas that we could address but won't for lack of space.

We have done this earlier with sensing [20]. Likewise we want to mention here that the University of Chicago is running a seven week Professional Certificate in Quantum Science, Networking and Communication this Fall [16]. They had another one on Quantum Engineering in general two years ago and they are planning others.

4.1 Workshops for High-School Teachers

There are several excellent outreach groups that the reader needs to be aware of and we mention here the University of Waterloo's Institute for Quantum Computation Schrödinger's Class [11] and the numerous workshops for HS teachers organized by Quantum for All [32]. By extension, relevant here: [26], [31] and [27].

5 QUANTUM ENGINEERING

5.1 From Misty States to Linear Algebra

Now one can make the transition like Diana Franklin in EdX (her first course [18] is in Terry Rudolph's terms up and including Bernstein-Vazirani). Her second course [19] starts from that and brilliantly seamlessly moves into using linear algebra. The many available exercises (including Qiskit exercises) with instant feedback are quintessential to the success of her commendable approach.

5.1.1 List of Topics. This is the list of topics we propose for a one semester class that could be extended to a two-semester sequence if supported by an adequate number of lab sessions:

- The Wave-Particle Duality Principle.
- The Uncertainty Principle in the Double-Slit Experiment.
- Qubits. Superposition. Measurement. Photons as qubits.
- Basic probability, trigonometry, simple vector spaces.
- Supporting formalisms: complex numbers, Euler's formula.
- Systems of two qubits. Entanglement. Bell states.
- The No-Signaling theorem.
- Axioms of QM: the superposition principle, the measurement axiom, and the unitary evolution of quantum states.
- Single qubit gates: X, Z, H, etc.
- Two qubit gates and tensor products. Working with matrices.
- The No-Cloning Theorem.
- The Quantum Teleportation protocol.
- Early quantum algorithms: Deutsch-Josza, Bernstein-Vazirani.
- Simon's algorithm (as a precursor to Shor's algorithm)
- Deutsch-Josza with Mach-Zehnder Interferometers.
- Quantum Factoring (Shor's Algorithm)
- Quantum Search (Grover's Algorithm)
- Physical implementation of qubits.
- The nine qubit modalities currently in use.
- Classical control of a Quantum Processing Unit (QPU)
- Error mitigation and control. NISQ and beyond.
- Post-quantum encryption

- Quantum Key Distribution (QKD).
- The Quantum Internet.
- Adiabatic Quantum Computation (AQC)
- Quantum Annealing (QA)

Useful references here are [9, 22, 30, 50]. Each one of these references will support the list of topics above while pleasantly surprising the reader with additional material. For a hardware lab [45] in conjunction with Qiskit Metal would make a great combination.

5.1.2 Illustrative Learning Outcomes. With this list of topics in mind, at the end of the course we would want students to Understand that:

- a quantum object (a) is produced as a particle, (b) propagates like a wave, and (c) is detected as a particle with a probability distribution that corresponds to a wave
- at the quantum level nature is inherently probabilistic.
- entanglement can be used to create non-classical correlations, but there is no way to use quantum entanglement to send messages faster than the speed of light.
- nature is inconsistent with any local hidden variable theory.
- quantum gates implement time evolution of a quantum state.

Become aware of the following:

- the power and idiosyncrasies of quantum communication
- the power of quantum parallelism and the role of constructive vs destructive interference in quantum algorithms given the probabilistic nature of measurement(s).

Understand that:

- quantum computation breaks the extended Church-Turing thesis but does not violate the original Church-Turing thesis and what the difference is
- quantum computation already occurs in nature. We are just trying to get better at harnessing it

Understand:

- the role of quantum Fourier sampling and quantum Fourier transform (QFT) in Shor's algorithm
- the classical components/aspects in Shor's algorithm
- the mechanisms of phase inversion and inversion around the mean in Grover's algorithm

Be able to:

- enumerate, compare and contrast the implementation-level specifics of each qubit modality (e.g., trapped ion, superconducting, silicon spin, photonic, quantum dot, neutral atom, topological, color center, electron-on-helium, etc.)
- pinpoint differences between adiabatic quantum computing (AQC, QA) and the gate model of quantum computation and which kind of problems each is better suited to solve

Understand that:

- a QPU is a heterogeneous multicore architecture, similarly to a FPGA or a GPU
- the building blocks of a quantum computer are: a quantum algorithm, a quantum language, a compiler, arithmetic, instruction set, micro-architecture, a quantum to classical conversion and a quantum chip.

5.2 Quantum Processing Unit (QPU)

The promise of quantum computers stems not from them being a conventional computer "killer", but rather from their ability to dramatically extend the kinds of problems that are tractable within computing. There are important computational problems that are easily calculable on a quantum computer, but that would be impossible on any conceivable standard computing device we could ever hope to build. But crucially, these speedups have only been seen for certain problems. Although it is anticipated that more will be discovered, it is highly unlikely that it would ever make sense to run all computations on a quantum computer. A quantum computer performs no better for most of the tasks taking up our laptop's clock cycles. Thus, a quantum computer is a co-processor from the programmer's viewpoint.

In the past, computers have used various co-processors, each suited to their specialities, such as floating-point arithmetic, signal processing, and real-time graphics. As with other co-processors such as the GPU (Graphics Processing Unit), programming for a QPU involves the programmer writing code that will primarily run on a typical computer's CPU (Central Processing Unit). The CPU only issues the QPU co-processor commands to initiate tasks suited to its capabilities. In [28, 29] PsiQuantum's Eric R Johnston enumerates the following distinguishing features for what it is like to program for a QPU:

- It is very rare that a program will run entirely on a QPU. Usually, a program running on a CPU will issue QPU instructions and later retrieve the results.
- Some tasks are very well suited to the QPU, while others are not.
- The QPU runs on a separate clock from the CPU and usually has its dedicated hardware interfaces to external devices (such as optical outputs).
- A typical QPU has its particular RAM, which the CPU cannot efficiently access.
- A simple QPU will be one chip accessed by a laptop, or even perhaps eventually an area within another chip. A more advanced QPU is a large, expensive add-on and requires special cooling.
- Early QPUs, even simple ones, are the size of refrigerators and need unique high-amperage power outlets.
- When a computation is done, a projection of the result is returned to the CPU, and most of the QPU's internal working data is discarded. QPU debugging can be very tricky, requiring special tools/techniques. Stepping through a program can be difficult, and often the best approach is to make changes to the program and observe their effect on the output.
- Optimizations speeding up one QPU may slow down another.

Though this list is somewhat daunting, one can replace QPU with GPU in every one of those statements, which are still entirely valid. Although QPUs are an almost alien technology of incredible power when it comes to the problems, we might face in learning to program them, a generation of software engineers have seen it all before. It is true, of course, that there are some nuances to QPU programming that are genuinely novel, but the uncanny number of similarities should be, overall, reassuring.

QPUs differ significantly (in some respects) from one qubit modality to another. Still, a significant amount of classical control is transferrable from one qubit modality to another.

5.3 Implementing Qubits

We acknowledge nine current modalities [36, 37]: superconducting, silicon spin, optical (photonics), quantum dots, trapped ions, color centers in diamond, neutral atoms in an optical tweezer array, topological and electron on helium.

5.4 Error Mitigation and Control

The complexity of noise is a crucial issue. Quantum error correction protects quantum information from errors due to decoherence and other quantum noise. Quantum error correction is theorised as essential to achieving fault-tolerant quantum computation that can reduce the effects of noise on stored quantum information, faulty quantum gates, faulty quantum preparation, and inaccurate measurements.

Classical error correction employs redundancy. The simplest albeit inefficient approach is the repetition code. The idea is to store the information multiple times, and—if these copies are later found to disagree—take a majority vote; e.g. suppose we copy a bit in the one state three times, etc. However, copying quantum information is impossible due to the no-cloning theorem. This theorem seems to present an obstacle to formulating a theory of quantum error correction. But it is possible to spread the (logical) information of one qubit onto a highly entangled state of several (physical) qubits. Peter Shor first discovered this method of formulating a quantum error correcting code by storing the information of one qubit onto a highly entangled state of nine qubits.

6 CONCLUSIONS

Interest in incorporating quantum architecture topics in the traditional CS curriculum remains high [38]. In this paper, we have argued that there is more than one way to achieve that goal, and we reviewed many available resources. And while none of our questions admits a clear-cut answer, we can try to summarize here our position as follows:

(1) the CS undergraduate should have a proper appreciation for the quantum mechanical nature of our world. They should know that there is more than one way to implement a qubit, that we are currently in the NISQ era, and there is a gate model as well as an alternative, adiabatic model of quantum computation.
(2) there are many entry points in such a program and consequently an equal number of associated prerequisites. The main prerequisite should be a certain intellectual versatility, manifested in a willingness to be exposed to information from more than one domain/discipline.
(3) in quantum computation labs will be quintessential, and they will rely on computer-assisted mathematics (e.g., Wolfram Alpha, `numpy`, Qiskit, `matplotlib`, etc.) software emulation (Qiskit Metal) traditional maths (Google Colab and LaTeX) access to actual quantum computers via various cloud platforms (Amazon Braket, IBM Q, Xanadu Borealis, etc.) and occasionally access to a physics lab, fab or foundry.
(4) a genuine interdisciplinary program can only be built if faculty has wide general support towards such a goal. Cross-campus, inter-departmental communication and cooperation may not be trivial, and faculty need to know that they might have to make a concerted effort to achieve such a beneficial desiderate.
(5) incorporating material about all qubit modalities in the curriculum will ensure the material will remain relevant over a reasonably long period. Some qubit modalities (e.g., photons via bulk optics) might allow more accessible experimental setups than others (trapped ions or superconducting qubits). But the widespread opinion is that students should be exposed to more than one qubit modality, including the design and implementation of qubits (e.g., via Qiskit Metal) and error mitigation and (classical) control.

We live in an era of great promise and consequently a proportionate amount of confusion. More than 60 companies worldwide are building quantum computers at this writing. And while some companies have already started announcing publicly their goals of reaching a million qubits [49] by the end of the decade, one needs to keep in mind that it is normal for a company to have different positions when talking to investors and customers. There is clearly no doubt that there has been significant progress, and with that comes a certain amount of overhype.

But there is also a certain amount of underhype that goes underreported. Witness for example the following interview [13]with David Deutsch in The Economist:

> “Last year I saw their ion-trap experiment, where they were experimenting on a single calcium atom. The idea of not just accessing but manipulating it, in incredibly subtle ways, is something I totally assumed would never happen. Now they do it routinely. [And it] works in a completely different way that cannot be expressed classically. This is a fundamentally new way of harnessing nature. To me, it’s secondary how fast it is.”

We need to keep an open mind and prepare our students for all their possible futures [44].

ACKNOWLEDGMENTS

This work was done as part of CS2023: ACM/IEEE-CS/AAAI Computer Science Curricula (csed.acm.org). We thank the other members of the Architecture and Organization (AR) Knowledge Area Committee for feedback and support. Thanks also to Prof. Anthony Laing from the Quantum Engineering Technology center at the Univ of Bristol and Michael Biercuk from the Univ of Sydney and Q-CTRL as well as Zlatko Minev and Thomas George McConkey from IBM Quantum for conversations on some of the topics discussed in this paper. Doug Finke and Charles Robinson from the QED-C Workforce TAC helped us calibrate the survey and collect the input from the QED-C members. QED-C is a broad international group of stakeholders from industry, academia, national labs and professional organizations that aims to enable and grow the quantum industry and its associated supply chain. QED-C was established with support from NIST as part of the federal strategy for advancing QIST as per the National Quantum Initiative Act in 2018.

REFERENCES

[1] Scott Aaronson. 2013. *Quantum Mechanics since Democritus* (1st ed.). Cambridge University Press, Cambridge, UK.

[2] Abraham Asfaw, Alexandre Blais, Kenneth R. Brown, Jonathan Candelaria, Christopher Cantwell, Lincoln D. Carr, Joshua Combes, Dripto M. Debroy, John M. Donohue, Sophia E. Economou, Emily Edwards, Michael F. J. Fox, Steven M. Girvin, Alan Ho, Hilary M. Hurst, Zubin Jacob, Blake R. Johnson, Ezekiel Johnston-Halperin, Robert Joynt, Eliot Kapit, Judith Klein-Seetharaman, Martin Laforest, H. J. Lewandowski, Theresa W. Lynn, Corey Rae H. McRae, Celia Merzbacher, Spyridon Michalakis, Prineha Narang, William D. Oliver, Jens Palsberg, David P. Pappas, Michael G. Raymer, David J. Reilly, Mark Saffman, Thomas A. Searles, Jeffrey H. Shapiro, and Chandralekha Singh. 2022. Building a Quantum Engineering Undergraduate Program. *IEEE Trans. Educ.* 65, 2 (2022), 220–242. https://doi.org/10.1109/TE.2022.3144943

[3] Edwin Barnes. 2022. *CSAAPT 2022 Presentation: Teaching quantum information science to high-school (and early undergraduates).* Retrieved August 18, 2022 from https://indico.phys.vt.edu/event/48/contributions/996/attachments/786/1083/CSAAPT_2022_Barnes.pdf

[4] Michael J Biercuk. 2016. *Building the Quantum Future (TEDxSydney).* Retrieved August 18, 2022 from https://www.youtube.com/watch?v=HS_BUd0rh_A

[5] Michael J Biercuk. 2021. *Quantum Technology Masters Degrees from an Industry Perspective.* Retrieved August 18, 2022 from https://www.linkedin.com/pulse/quantum-technology-masters-degrees-from-industry-michael-biercuk/

[6] Yuly Billig. 2018. *Quantum Computing for High School Students* (1st. ed.). Qubit Publishing, Ottawa, Ontario, Canada.

[7] Yuly Billig. 2018. *Quantum Computing for High-School Students (electronic version in portable document format).* Retrieved August 18, 2022 from https://www.qubitpublishing.com/gallery/qc-high-2e-with-cover.pdf

[8] Reinhold Blumel. 2010. *Foundations of Quantum Mechanics: From Photons to Quantum Computers.* Jones and Bartlett, Sudbury, MA.

[9] Mahn-Soo Choi. 2022. *A Quantum Computation Workbook)* (1st ed.). Springer Nature, Cham, Switzerland.

[10] Ivan B Djordjevic. 2010. *Quantum Information Processing, Quantum Computing, and Quantum Error Correction (An Engineering Approach)* (2nd ed.). Academic Press, Elsevier, Cambridge, MA, USA.

[11] John Donohue. 2022. *Institute for Quantum Computing at the University of Waterloo.* Retrieved August 19, 2022 from https://uwaterloo.ca/institute-for-quantum-computing/outreach-and-workshops

[12] Andrew S. Dzurak, Julien Epps, Arne Laucht, Robert Malaney, Andrea Morello, Hendra I. Nurdin, Jarryd J. Pla, Andre Saraiva, and Chih Hwan Yang. 2022. Development of an Undergraduate Quantum Engineering Degree. *IEEE Transactions on Quantum Engineering* 3 (2022), 1–10. https://doi.org/10.1109/TQE.2022.3157338

[13] The Economist. 2017. *David Deutsch, father of quantum computing.* Retrieved August 18, 2022 from https://www.economist.com/technology-quarterly/2017/03/09/david-deutsch-father-of-quantum-computing

[14] Sophia Economou, Terry Rudolph, and Edwin Barnes. 2020. *Teaching quantum information science to high-school and early undergraduate students.* Retrieved August 18, 2022 from https://arxiv.org/abs/2005.07874

[15] Emily Edwards, James Freericks, Carol Lynn Alpert, and Diana Franklin. 2020. *Key Concepts for Future QIS Learners.* Retrieved August 17, 2022 from https://files.webservices.illinois.edu/9156/keyconceptsforfutureqislearners5-20.pdf

[16] Chicago Quantum Exchange. 2022. *Certificate in Quantum Science, Networking, and Communications.* Retrieved August 19, 2022 from https://professional.uchicago.edu/find-your-fit/courses/quantum-science-networking-and-communications

[17] Michael D Fayer. 2010. *Absolutely Small: How Quantum Theory Explains Our Everyday World* (1st. ed.). AMACOM, New York, NY.

[18] Diana Franklin. 2021. *Quantum Computing for Everyone (Part 1).* Retrieved August 19, 2022 from https://learning.edx.org/course/course-v1:UChicagoX+QUAN11000+3T2021/home

[19] Diana Franklin. 2021. *Quantum Computing for Everyone (Part 2).* Retrieved August 19, 2022 from https://learning.edx.org/course/course-v1:UChicagoX+QUAN12000+2T2021/home

[20] James K Freericks. 2021. *GeorgetownX Professional Certificate in Quantum Sensing.* Retrieved August 16, 2022 from https://www.edx.org/professional-certificate/georgetownx-foundations-of-quantum-sensing

[21] Adrian German. 2021. *In-depth Review of Georgetown's Quantum Mechanics Course: A comprehensive quantum mechanics course with an engaging instructor and a concepts-first approach.* Retrieved August 16, 2022 from https://www.classcentral.com/report/review-quantum-mechanics

[22] Jack D Hidary. 2021. *Quantum Computing: An Applied Approach* (2nd ed.). Springer Nature, Cham, Switzerland.

[23] John Horgan. 2016. *Scott Aaronson Answers Every Ridiculously Big Question I Throw at Him (Scientific American).* Retrieved August 19, 2022 from https://blogs.scientificamerican.com/cross-check/scott-aaronson-answers-every-ridiculously-big-question-i-throw-at-him/

[24] Ciaran Hughes, Joshua Isaacson, Anastasia Perry, Ranbel F Sun, and Jessica Turner. 2021. *Quantum Computing for the Quantum Curious* (1st ed.). Springer, Cham, Switzerland.

[25] Ciaran Hughes, Joshua Isaacson, Anastasia Perry, Ranbel F Sun, and Jessica Turner. 2021. *Quantum Computing for the Quantum Curious.* Retrieved August 18, 2022 from https://link.springer.com/book/10.1007/978-3-030-61601-4

[26] Peter Shor Isaac Chuang, Will Oliver. 2017. *Quantum Computing (Fundamentals).* Retrieved October 5, 2022 from https://learn-xpro.mit.edu/quantum-computing

[27] Peter Shor Isaac Chuang, Will Oliver. 2017. *Quantum Computing (Realities).* Retrieved October 5, 2022 from https://prolearn.mit.edu/quantum-computing-realities

[28] Eric R Johnston. 2020. *Anatomy of a Quantum Processing Unit (as part of a SIGCSE 2020 full day pre-symposium event entitled Programming Quantum Computers: Tools and Techniques for the CSCI Undergraduate Faculty).* Retrieved August 19, 2022 from https://legacy.cs.indiana.edu/~dgerman/quantum-computing-for-undergrads/videos.html

[29] Eric R Johnston, Nic Harrigan, and Mercedes Gimeno-Segovia. 2019. *Programming Quantum Computers: Essential Algorithms and Code Samples* (1st ed.). O'Reilly Media, Sebastobol, CA, USA.

[30] Ray LaPierre. 2021. *Introduction to Quantum Computing* (1st ed.). Springer Nature (The Materials Research Society Series), Cham, Switzerland.

[31] Araceli Venegas-Gomez Maninder Kaur. 2022. *Defining the quantum workforce landscape: a review of global quantum education initiatives.* Retrieved October 5, 2022 from https://arxiv.org/abs/2202.08940

[32] Karen Jo Matsler. 2022. *Quantum for All.* Retrieved August 19, 2022 from https://quantumforall.org/

[33] David A B Miller. 2008. *Quantum Mechanics for Scientists and Engineers* (1st. ed.). Cambridge University Press, Cambridge, UK.

[34] Richard A Muller. 2010. *Physics and Technology for Future Presidents (An Introduction to the Essential Physics Every World Leader Needs to Know* (1st ed.). Princeton University Press, Princeton, NJ, USA.

[35] Industry NSF Interagency Working Group on Workforce and Infrastructure. 2020. *Quantum Information Science Learning: Future Pathways.* Retrieved August 17, 2022 from https://qis-learners.research.illinois.edu/about/

[36] Dominic Walliman (Domain of Science). 2022. *The Map of Quantum Computing.* Retrieved August 19, 2022 from https://www.youtube.com/watch?v=-UlxHPIEVqA

[37] Dominic Walliman (Domain of Science). 2022. *Who Has The Best Quantum Computer?* Retrieved August 19, 2022 from https://www.youtube.com/watch?v=gcbMKt079l8

[38] Marcelo Pias, Brett Becker, Qiao Xiang, Mohamed Zahran, and Monica Anderson. 2022. Should Quantum Processor Design be Considered a Topic in Computer Architecture Education? *SIGCSE 2022: Proceedings of the 53rd ACM Technical Symposium on Computer Science Education* (March 2022). https://doi.org/10.1145/3478432.3499201

[39] Michael G Raymer. 2017. *Quantum Physics: What Everyone Needs to Know* (1st. ed.). Oxford University Press, New York, NY.

[40] Eleanor G Rieffel and Wolfgang H Polak. 2000. *An Introduction to Quantum Computing for Non-Physicists.* Retrieved August 19, 2022 from https://arxiv.org/abs/quant-ph/9809016

[41] Eleanor G Rieffel and Wolfgang H Polak. 2014. *Quantum Computing: A Gentle Introduction* (1st ed.). The MIT Press, Cambridge, MA, USA.

[42] Terry Rudolph. 2017. *Q is for Quantum* (1st. ed.). Terry Rudolph, Orlando, FL.

[43] Terry Rudolph. 2021. *Schedule of events for the QED-C Workforce TAC (see March 25 entry: Teaching quantum theory using only rules of basic arithmetic).* Retrieved August 18, 2022 from https://legacy.cs.indiana.edu/~dgerman/qed-c/wd-tac/schedule.html

[44] David Shaw. 2021. *Quantum Hardware Outlook 2022.* Retrieved August 18, 2022 from https://www.factbasedinsight.com/quantum-hardware-outlook-2022/

[45] Daniel D Stancil and Gregory T Byrd. 2022. *Principles of Superconducting Quantum Computers)* (1st ed.). John Wiley and Sons, Inc., Hoboken, NJ.

[46] Daniel F Styer. 2004. *The Strange World of Quantum Mechanics* (1st. ed.). Cambridge University Press, Cambridge, UK.

[47] Daniel F Styer. 2022. *Invitation to Quantum Mechanics* (1st. ed.). World Scientific, Singapore.

[48] Wilton L Virgo. 2014. *Quantum Mechanics in Everyday Life* (1st ed.). Prentice Hall PTR, Upper Saddle River, NJ, USA.

[49] John Russell (HPC Wire). 2022. *PsiQuantum's Path to 1 Million Qubits.* Retrieved April 21, 2022 from https://www.hpcwire.com/2022/04/21/psiquantums-path-to-1-million-qubits-by-the-middle-of-the-decade/

[50] Hiu Yung Wong. 2022. *Introduction to Quantum Computing (From a Layperson to a Programmer in 30 Steps)* (1st ed.). Springer Nature, Cham, Switzerland.

[51] M Suhail Zubairy. 2020. *Quantum Mechanics for Beginners (with applications to quantum communication and quantum computing)* (1st. ed.). Oxford University Press, Oxford, UK.