Quantum Computing

Program Next-Gen Computers for Hard, Real-World Applications

The Pragmatic Bookshelf
Raleigh, North Carolina

The Pragmatic Programmers

Quantum Computing

Program Next-Gen Computers for Hard, Real-World Applications

Nihal Mehta, Ph.D.

edited by Brian MacDonald

Quantum Computing

Program Next-Gen Computers for Hard, Real-World Applications

Nihal Mehta, Ph.D.

The Pragmatic Bookshelf
Raleigh, North Carolina

The team that produced this book includes:

Publisher: Andy Hunt
VP of Operations: Janet Furlow
Executive Editor: Dave Rankin
Development Editor: Brian MacDonald
Copy Editor: L. Sakhi MacMillan
Indexing: Potomac Indexing, LLC
Layout: Gilson Graphics

For sales, volume licensing, and support, please contact *support@pragprog.com*.

For international rights, please contact *rights@pragprog.com*.

ISBN-13: 978-1-68050-720-1
Encoded using the finest acid-free high-entropy binary digits.
Book version: P1.0—August 2020

Preface

Richard Feynman, a Nobel Laureate, first postulated[1] a computer rooted in the laws of quantum mechanics that govern the behavior of subatomic particles such as muons, gluons, quarks, and bosons. It's a theory that strains credulity at every turn—its bizarre concepts jar against everyday experience. A long line of distinguished quantum physicists from Niels Bohr to Albert Einstein and Richard Feynman have been exasperated by the very theory they helped create. Yet, surprisingly, the theory has correctly predicted every phenomenon that physicists have thrown at it. These same principles now have their sights trained on digital computing. For the first time since Eniac, the modern computer built at the University of Pennsylvania in 1946, the underlying fabric of computing is threatened: bits, with their rigid `0` and `1` states, are replaced with *fluid* units that seemingly exist in both states simultaneously, giving us unprecedented ways to tackle challenging computational tasks. Quantum mechanics is the most spectacular theory ever put forth, and we are going to latch onto its coattails to learn about quantum computing and how it's challenging the notions of conventional computing.

I first got interested in quantum computing years ago when I was exploring ways to squeeze better performance from the mathematical-based computer models that I was tasked with building for some of the world's largest organizations. Time and again, though the computer models would return acceptable solutions, I couldn't get them to do even better. It's like the models hit a wall. Quantum mechanical concepts such as tunneling seemed to offer a way to punch through the barrier. But quantum mechanics is strange, and it was far from clear how to model computational tasks using subatomic particles. I created "paper" models to help make sense of the counter-intuitive quantum mechanics concepts. But, with no real quantum hardware to validate the models, they remained untested ideas that I filed away.

1. https://link.springer.com/article/10.1007/BF02650179

Over the years, there's been much research[2] on testing out small pieces of quantum machinery, but there was no integrated computer on which one could write any meaningful code. All this changed in 2017 when IBM released a quantum computer which, incredibly, was freely available to one and all—quantum computing was no longer the sole province of high-octane research laboratories. Armed with an internet connection, anyone could get their hands on some über-cool computer technology. I was finally able to test and refine my "paper" models on an actual quantum computer.

But writing a book on quantum computing poses a singular challenge of weaving a coherent tale from two distinct threads: its origins in quantum mechanics and its relationship with computer science. As a result, the tools of one are used to explain the other—which, for computer science, boils down to treating the inherent random nature of quantum computing as a branch of probability theory. Neither of these approaches is satisfactory. It takes a fair amount of effort, for example, to build the intuition for probabilistic algorithms, and at the same time, quantum mechanics is alien to many computer professionals.

I, however, believe that there's a third aspect that's overlooked: a conceptual bridge that straddles both quantum mechanics and computer science. I introduce a way to model quantum phenomena based on quantum mechanical principles that intersects computer science precepts and gives a way to master the big ideas of quantum computing. This will help you better appreciate why it's garnering attention and why computer scientists and corporations think it's the next big bet and are willing to spend big dollars. Mathematics underpins much of quantum mechanics and is essential in any investigation of quantum computing. But, with this model, I promise to not use mathematics as a refuge to get to the bottom of thorny topics. I'll factor the fundamental concepts of quantum mechanics on the computer science base layer in a natural and intuitive way that neatly brings in the quantum mechanical concepts without baffling you. You'll begin to see these models from Chapter 2, Goodbye Mr. Bits—From Classical to Quantum Bits, on page ? onward.

Is This Book for Me?

This book is for developers new to quantum programming as well as those who may have read and heard about this technology and are looking for a quick way to get started. This book will also be helpful to students who are studying quantum computing at university. They'll find that the topics covered in this book complement their classroom lectures.

2. https://en.wikipedia.org/wiki/Timeline_of_quantum_computing

I don't expect you to have any background or previous exposure to quantum mechanics or quantum programming. I'll introduce you to the relevant concepts and help you gain the necessary dexterity. But I do expect that you know how to program using any one of the multitude of high-level languages available today, such as C#, Java, JavaScript, Objective-C, Python, and so on. Although you don't need to know assembly language programming, you should at least be familiar with the notion of Boolean gates and logical operations, and the fundamentals of complex numbers, basic trigonometry, and vector and matrix operations, such as multiplying them and finding their inverses. (See Appendix 1, Mathematical Review, on page ?, to brush up on these subjects.) You'll need to know these topics to apply advanced quantum effects in your programs from Chapter 6, Designer Genes—Custom Quantum States, on page ?, onward.

After reading this book, you'll learn the following:

Introducing quantum effects in your programs
: You'll understand the value that quantum mechanics can bring to solving computational tasks over that of classical machines. You'll learn a new mindset to manipulate bits using quantum principles.

Apply quantum computing to real world problems
: You'll design quantum algorithms to solve real world problems. One of the major themes of this book is to develop your intuition so that you properly apply quantum concepts for your own challenging computational tasks.

Discern which problems are suitable for quantum computers
: Not all applications are suited for quantum computers. You'll learn to identify the problems that are best suited for quantum computers and which ones are better solved using conventional computers.

Interface with a quantum computer from within your application
: By invoking a quantum computer from within your own program, you can execute the heavy-lifting portion of your code on a quantum computer and transparently return results back to your application.

How Will This Book Give Me What I Want?

Quantum mechanics can be an imposing hurdle to get over before becoming proficient in quantum computing. Yet a systematic exploration of the fundamental ideas can make this subject engaging. We take a hands-on-first approach in this book by emphasizing writing programs on an actual quantum computer so that you get a concrete handle on the quantum mechanical concepts.

I've organized the material using familiar ideas from classical computers in *Goodbye Mr. Bits—From Classical to Quantum Bits* through *Beam Me Up, Scotty—Quantum Tagging and Entangling*. You'll go to the heart of quantum programming and get a taste for how quantum effects are reshaping computing. In subsequent chapters, you'll learn to tailor quantum phenomena in your own quantum programs.

- We begin in Chapter 1, Hello Quantum, on page ?, by demonstrating how quantum computing is trespassing on the traditional turf of conventional computing. Learning quantum programming without using a motivating application is like trying to explore a new city by riding in the subway. The subway, like programming syntax, can take you from one stop to another, but you have to step out to experience the city's sights and culture. Thus, you'll set up and run a quantum program on a standard scheduling problem and see for yourself how a quantum computer solves it.

- The traditional way of analyzing computers with `0` and `1` bits seems to not apply with quantum computing. In Chapter 2, Goodbye Mr. Bits—From Classical to Quantum Bits, on page ?, you'll be introduced to a way to think about quantum computing using the standard tools of classical computing. You'll learn about quantum bits and compare and contrast them with the binary bits of conventional computing. You'll see that quantum bits exist in a blended state of the two binary bits. You'll also learn the *Qubelets Model*, a new way to think about quantum programming that computer professionals will find familiar. Much like Google's Street View that lets you explore a city from your desk, the model you'll get to work with rapidly gets to the heart of quantum computing using intuitive concepts. These ideas animate the subsequent quantum mechanical principles used for computation in the later chapters.

- In Chapter 3, Elementary, My Dear Watson—Quantum Logic, on page ?, you'll work with quantum gates that are largely similar to their classical counterparts. These gates build the scaffolding for the parts where the quantum mechanical effects take place in your program. By themselves, the quantum logic gates don't offer any inherent advantages over their classical counterparts. You can build quantum programs with these gates, but your code won't perform better than that designed to work on classical computers.

- In Chapter 4, All Together Now—Quantum Superposition, on page ?, and Chapter 5, Beam Me Up, Scotty—Quantum Tagging and Entangling, on page ?, you'll learn why quantum computing has the potential to corner the market for solving hard computational problems. These quantum mechanical concepts constitute the core of quantum computing applications. You'll learn about quantum gates that allow you to nudge quantum bits around in ways you can't with classical computers.

Quantum computing isn't fantasy but is solidly grounded in reality. To this end, we'll show that the Qubelets Model explains the quantum mechanical nature of a well-known set of physics experiments and illustrate its connections with quantum programming. You'll be introduced to the Mega-Qubit on page ?—a framework that'll help you understand and work with quantum superposition, a quantum phenomenon that manages all possible solutions at once.

- In Chapter 6, Designer Genes—Custom Quantum States, on page ?, Chapter 7, Small Step for Man—Single Qubit Programs, on page ?, and Chapter 8, Giant Leap for Mankind—Multi-Qubit Programs, on page ?, you'll learn to design quantum algorithms where you can precisely control quantum bits to suit your purpose. Understanding these concepts is crucial to writing quantum programs that work reliably.

  There's a handy list of gates, which groups the different ways to introduce quantum effects when designing your quantum algorithms for your own problems.

- In Chapter 9, Alice in Quantumland—Quantum Cryptography, on page ?, and Chapter 10, Quantum Search, on page ?, you'll learn how the core quantum concepts of *entanglement* and *superposition* are used in real-world applications. You'll learn how quantum computing encrypts virtually break-proof messages and also about algorithms that sweep through the potential solutions of hard-to-solve computational tasks rapidly.

- While learning quantum computing, it's easy to forget its origin in quantum mechanics and the oddball principles that govern the natural world around us. So if you're interested to know more about how quantum computing is strongly intertwined with quantum mechanics, see Appendix 3, Quantum Mechanics with Qubelets, on page ?, to get a peek at the deep relationship between them.

Most chapters have programming exercises and problems that reinforce concepts you've learned in the chapter and help you become familiar with this new technology. All exercises and problems have solutions. So you'll find them instructive even if you glance through them.

If You Want Just the Basics

If you'd like to learn just the basic ideas of quantum computing, largely avoiding complex numbers and trigonometry, read the following:

- Chapter 1, Hello Quantum, on page ?, through Chapter 5, Beam Me Up, Scotty—Quantum Tagging and Entangling, on page ?.
- Quantum States and Probabilities, on page ?, in Chapter 6, Designer Genes—Custom Quantum States, on page ?.
- Quantum States as Vectors, on page ?, Quantum Gates as Matrices, on page ?, and Sequence of Gates as Matrix Multiplication, on page ?, in Chapter 7, Small Step for Man—Single Qubit Programs, on page ?.
- Chapter 8, Giant Leap for Mankind—Multi-Qubit Programs, on page ?, onward.

If You're Impatient

If you're in a hurry to learn about quantum effects, you can jump ahead to Chapter 4, All Together Now—Quantum Superposition, on page ?, and see how quantum computers handle all possible states of a computational problem.

If You're Really, Really Impatient

If quantum computing is gnawing at you and you're already somewhat familiar with quantum mechanics, dive straight to Chapter 6, Designer Genes—Custom Quantum States, on page ?. When using quantum computers to solve hard problems modeled with multiple qubits, you'll see that the concepts introduced in Qubelets Model, on page ?, are an alternative to using the standard Bloch sphere. You'll learn that qubelets let you visualize how qubits interact with each other and figure out where to introduce quantum effects. You'll learn how to precisely create tailor-made quantum states in your programs and then carefully fine-tune them to solve your computational problem.

What's Unique in This Book?

If you're new to quantum computing, you'll find that the material in this book is quite different from what you'll see in others. In fact, it's only in Chapter 6, Designer Genes—Custom Quantum States, on page ?, that you'll begin to see parallels with the literature.

If you're already familiar with quantum computing, you'll find a new, and hopefully more intuitive, take on standard concepts. In my mind, the following sections, in particular, stand out most:

Modeling Quantum Bits with the Qubelets Model, on page ?: Describes a way to think about quantum states as components similar to, say, analyzing a block on an inclined plane in classical physics. Specifically, we introduce the *qubelets* to model quantum states.

Multi-Qubit Superposition: The Mega-Qubit, on page ?: Getting a handle on quantum superposition using the conceptual device of the *mega-qubit*.

Intuition Behind Entanglement, on page ?: Making sense of *quantum entanglement* with qubelets and the mega-qubit.

Rotating Qubelets Through Any Angle, on page ?: Relating the standard Bloch sphere prevalent in quantum computing with the qubelets introduced in this book.

Classifying Quantum Gates, on page ?: New way to categorize quantum gates, based on how they directly affect qubits rather than on abstract rotations around the Bloch sphere.

Teleporting Mega-Qubit, on page ?: Seeing the teleporting states as a concrete concept rather than as esoteric calculations.

Tell-Your-Boss Version: The "Key" Idea, on page ?: Thinking of quantum cryptography as a way to send many messages on the same channel at the same time to confuse anyone illegally listening in.

Canceling Circuit, on page ?: Explains how to control the odds of a quantum program returning the correct result.

Online Resources

All source code for the examples in this book can be downloaded from the book's page on the Pragmatic Bookshelf's website at https://pragprog.com/book/nmquantum/quantum-computing. You'll also find an errata page at https://pragprog.com/book/nmquantum/errata.

Acknowledgments

I'm grateful that the Pragmatic Bookshelf agreed to publish this book. The current technical-books landscape is teeming with so many books that it's hard to discern any publisher's unique imprint. But, from their very earliest books, I've been struck by the winsome style of Pragmatic books. Although the credit for producing such well-written books lies with the respective authors, there seemed to be something else at play too—a secret style guide of some sort that makes every Pragmatic book a delight to read. I wanted my book to have that same secret sauce.

I struck gold with Brian MacDonald as the Development Editor. Right from the start, Brian understood the challenges of making this particular topic presentable. His comments and guidance were crucial to ensuring that I wrote the material in a way that the reader would find useful, without devolving into minutiae that is so easy to fall into with technical content. It was also Brian's suggestion to start the book with a practical example that immediately demonstrates quantum computing instead of subjecting the reader to abstract theory and relying on the reader's patience to plow through before seeing something useful. I believe that you'll enjoy the book much more as a result of this approach.

I deeply appreciate the efforts of Susan Conant, L. Sakhi MacMillan, and all the other editors at Pragmatic Bookshelf for helping shape this book. They always reviewed the manuscript promptly, and their advice raised the quality of the presentation.

I want to thank Meghan Blanchette, who was the first editor on this book. Meghan gave me, a first-time author, advice on tightening up my writing that is reflected on every page.

I also want to thank the technical reviewers and all the readers that submitted errata. Their detailed comments and feedback encouraged me that I was on the right track. The technical reviewers for this book were: Craig Castelaz, Ashish Dixit, Jan Goyvaerts, Nigel Lowry, Nick McGinness, Tim Nugent, Kim Shrier, Tibor Simic, Gianluigi Spagnuolo, Charley Stran, Stefan Turalski, Nick Watts, Boris Vasile, Mitchell Volk, and Stephen Wolff. Two readers in particular, Kaoru Hosokawa and Michael Blake, identified several errata throughout the beta release cycle.

The impetus for writing this book came when it finally dawned on me in the summer of 2018 that my daughter, Mira, would be going away to the University of Chicago in the fall and I would need something to combat the loneliness that would set in.

Come, let's shine a beam of photons on the world of computer programming and see it in a new light.

Nihal Mehta

September 2019