N. David Mermin

Quantum Computer Science

An Introduction

CAMBRIDGE

Quantum Computer Science

An Introduction

N. David Mermin

Cornell University

CAMBRIDGE
UNIVERSITY PRESS

CAMBRIDGE UNIVERSITY PRESS
Cambridge, New York, Melbourne, Madrid, Cape Town, Singapore, São Paulo

Cambridge University Press
The Edinburgh Building, Cambridge CB2 8RU, UK

Published in the United States of America by Cambridge University Press, New York

www.cambridge.org
Information on this title: www.cambridge.org/9780521876582

First published 2007

Printed in the United Kingdom at the University Press, Cambridge

A catalog record for this publication is available from the British Library

ISBN 978-0-521-87658-2 hardback

Quantum Computer Science
An Introduction

In the 1990s it was realized that quantum physics has some spectacular applications in computer science. This book is a concise introduction to quantum computation, developing the basic elements of this new branch of computational theory without assuming any background in physics. It begins with a novel introduction to the quantum theory from a computer-science perspective. It illustrates the quantum-computational approach with several elementary examples of quantum speed-up, before moving to the major applications: Shor's factoring algorithm, Grover's search algorithm, and quantum error correction.

The book is intended primarily for computer scientists who know nothing about quantum theory but would like to learn the elements of quantum computation either out of curiosity about this new paradigm, or as a basis for further work in the subject. It will also be of interest to physicists who want to learn the theory of quantum computation, and to physicists and philosophers of science interested in quantum foundational issues. It evolved during six years of teaching the subject to undergraduates and graduate students in computer science, mathematics, engineering, and physics, at Cornell University.

N. David Mermin is Horace White Professor of Physics Emeritus at Cornell University. He has received the Lilienfeld Prize of the American Physical Society and the Klopsteg award of the American Association of Physics Teachers. He is a member of the U.S. National Academy of Sciences and the American Academy of Arts and Sciences. Professor Mermin has written on quantum foundational issues for several decades, and is known for the clarity and wit of his scientific writings. Among his other books are *Solid State Physics* (with N. W. Ashcroft, Thomson Learning 1976), *Boojums all the Way Through* (Cambridge University Press 1990), and *It's about Time: Understanding Einstein's Relativity* (Princeton University Press 2005).

"This is one of the finest books in the rapidly growing field of quantum information. Almost every page contains a unique insight or a novel interpretation. David Mermin has once again demonstrated his legendary pedagogical skills to produce a classic."

Lov Grover, Bell Labs

"Mermin's book will be a standard for instruction and reference for years to come. He has carefully selected, from the mountain of knowledge accumulated in the last 20 years of research in quantum information theory, a manageable, coherent subset that constitutes a complete undergraduate course. While selective, it is in no sense "watered down"; Mermin moves unflinchingly through difficult arguments in the Shor algorithm, and in quantum error correction theory, providing invaluable diagrams, clear arguments, and, when necessary, extensive appendices to get the students successfully through to the end. The book is suffused with Mermin's unique knowledge of the history of modern physics, and has some of the most captivating writing to be found in a college textbook."

David DiVincenzo, IBM T. J. Watson Research Center

"Mermin's book is a gentle introduction to quantum computation especially aimed at an audience of computer scientists and mathematicians. It covers the basics of the field, explaining the material clearly and containing lots of examples. Mermin has always been an entertaining and comprehensible writer, and continues to be in this book. I expect it to become the definitive introduction to this material for non-physicists."

Peter Shor, Massachusetts Institute of Technology

"Textbook writers usually strive for a streamlined exposition, smoothing out the infelicities of thought and notation that plague any field's early development. Fortunately, David Mermin is too passionate and acute an observer of the cultural side of science to fall into this blandness. Instead of omitting infelicities, he explains and condemns them, at the same time using his experience of having taught the course many times to nip nascent misunderstandings in the bud. He celebrates the field's mongrel origin in a shotgun wedding between classical computer scientists, who thought they knew the laws of information, and quantum physicists, who thought information was not their job. Differences remain: we hear, for example, why physicists love the Dirac notation and mathematicians hate it. Worked-out examples and exercises familiarize students with the necessary algebraic manipulations, while Mermin's lucid prose and gentle humor cajole them toward a sound intuition for what it all means, not an easy task for a subject superficially so counterintuitive."

Charles Bennett, IBM T. J. Watson Research Center

Preface

It was almost three quarters of a century after the discovery of quantum mechanics, and half a century after the birth of information theory and the arrival of large-scale digital computation, that people finally realized that quantum physics profoundly alters the character of information processing and digital computation. For physicists this development offers an exquisitely different way of using and thinking about the quantum theory. For computer scientists it presents a surprising demonstration that the abstract structure of computation cannot be divorced from the physics governing the instrument that performs the computation. Quantum mechanics provides new computational paradigms that had not been imagined prior to the 1980s and whose power was not fully appreciated until the mid 1990s.

In writing this introduction to quantum computer science I have kept in mind readers from several disciplines. Primarily I am addressing computer scientists, electrical engineers, or mathematicians who may know little or nothing about quantum physics (or any other kind of physics) but who wish to acquire enough facility in the subject to be able to follow the new developments in quantum computation, judge for themselves how revolutionary they may be, and perhaps choose to participate in the further development of quantum computer science. Not the least of the surprising things about quantum computation is that remarkably little background in quantum mechanics has to be acquired to understand and work with its applications to information processing. Familiarity with a few fundamental facts about finite-dimensional vector spaces over the complex numbers (summarized and reviewed in Appendix A) is the only real prerequisite.

One of the secondary readerships I have in mind consists of physicists who, like myself – I am a theorist who has worked in statistical physics, solid-state physics, low-temperature physics, and mathematical physics – know very little about computer science, but would like to learn about this extraordinary new application of their discipline. I stress, however, that my subject is quantum computer science, not quantum computer design. This is a book about quantum computational software – not hardware. The difficult question of how one might actually build a quantum computer is beyond its scope.

Another secondary readership is made up of those philosophers and physicists who – again like myself – are puzzled by so-called foundational issues: what the strange quantum formalism implies about the nature of the world that it so accurately describes. By applying quantum mechanics in an entirely new way – and especially by applying it to the processing of knowledge – quantum computation gives a new perspective on interpretational questions. While I rarely address such matters explicitly, for purely pedagogical reasons my presentation is suffused with a perspective on the quantum theory that is very close to the venerable but recently much reviled Copenhagen interpretation. Those with a taste for such things may be startled to see how well quantum computation resonates with the Copenhagen point of view. Indeed, it had been my plan to call this book *Copenhagen Computation* until the excellent people at Cambridge University Press and my computer-scientist friends persuaded me that virtually no members of my primary readership would then have had any idea what it was about.

Several years ago I mentioned to a very distinguished theoretical physicist that I spent the first four lectures of a course in quantum computation giving an introduction to quantum mechanics for mathematically literate people who knew nothing about quantum mechanics, and quite possibly little if anything about physics. His immediate response was that any application of quantum mechanics that can be taught after only a four-hour introduction to the subject cannot have serious intellectual content. After all, he remarked, it takes any physicist many years to develop a feeling for quantum mechanics.

It's a good point. Nevertheless computer scientists and mathematicians with no background in physics have been able quickly to learn enough quantum mechanics to understand and make major contributions to the theory of quantum computation. There are two main reasons for this.

First of all, a quantum computer – or, more accurately, the abstract quantum computer that one hopes someday to be able to embody in actual hardware – is an extremely simple example of a physical system. It is discrete, not continuous. It is made up out of a finite number of units, each of which is the simplest possible kind of quantum-mechanical system, a so-called two-state system, whose behavior, as we shall see, is highly constrained and easily specified. Much of the analytical complexity of learning quantum mechanics is connected with mastering the description of continuous (infinite-state) systems. By restricting attention to collections of two-state systems (or even d-state systems for finite d) one can avoid much suffering. Of course one also loses much wisdom, but hardly any of it – at least at this stage of the art – is relevant to the basic theory of quantum computation.

Second, and just as important, the most difficult part of learning quantum mechanics is to get a good feeling for how the formalism

can be applied to actual phenomena. This almost invariably involves formulating oversimplified abstract models of real physical systems, to which the quantum formalism can then be applied. The best physicists have an extraordinary intuition for what features of the phenomena are essential and must be represented in a model, and what features are inessential and can be ignored. It takes years to develop such intuition. Some never do. The theory of quantum computation, however, is entirely concerned with an abstract model – the easy part of the problem.

To understand how to *build* a quantum computer, or even to study what physical systems are promising candidates for realizing such a device, you must indeed have many years of experience in quantum mechanics and its applications under your belt. But if you only want to know what such a device is capable in principle of doing once you have it, then there is no reason to get involved in the really difficult physics of the subject. Exactly the same thing holds for ordinary classical computers. One can be a masterful practitioner of computer science without having the foggiest notion of what a transistor is, not to mention how it works.

So while you should be warned that the subset of quantum mechanics you will acquire from this book is extremely focused and quite limited in its scope, you can also rest assured that it is neither oversimplified nor incomplete, when applied to the special task for which it is intended.

I might note that a third impediment to developing a good intuition for quantum physics is that in some ways the behavior implied by quantum mechanics is highly counterintuitive, if not downright weird. Glimpses of such strange behavior sometimes show up at the level of quantum computation. Indeed, for me one of the major appeals of quantum computation is that it affords a new conceptual arena for trying to come to a better understanding of quantum weirdness. When opportunities arise I will call attention to some of this strange behavior, rather than (as I easily could) letting it pass by unremarked upon and unnoticed.

The book evolved as notes for a course of 28 one-hour lectures on quantum computation that I gave six times between 2000 and 2006 to a diverse group of Cornell University undergraduates, graduate students, and faculty, in computer science, electrical engineering, mathematics, physics, and applied physics. With so broad an audience, little common knowledge could be assumed. My lecture notes, as well as my own understanding of the subject, repeatedly benefited from comments and questions in and after class, coming from a number of different perspectives. What made sense to one of my constituencies was often puzzling, absurd, or irritatingly simple-minded to others. This final form of my notes bears little resemblance to my earliest versions, having been improved by insightful remarks, suggestions, and complaints about everything from notation to number theory.

In addition to the 200 or so students who passed through P481-P681-CS483, I owe thanks to many others. Albert J. Sievers, then Director of Cornell's Laboratory of Atomic and Solid State Physics, started me thinking hard about quantum computation by asking me to put together a two-week set of introductory lectures for members of our laboratory, in the Fall of 1999. So many people showed up from all over the university that I decided it might be worth expanding this survey into a full course. I'm grateful to two Physics Department chairs, Peter Lepage and Saul Teukolsky, for letting me continue teaching that course for six straight years, and to the Computer Science Department chair, Charlie van Loan, for support, encouragement, and a steady stream of wonderful students. John Preskill, though he may not know it, taught me much of the subject from his superb online Caltech lecture notes. Charles Bennett first told me about quantum information processing, back when the term might not even have been coined, and he has always been available as a source of wisdom and clarification. Gilles Brassard has on many occasions supplied me with help from the computer-science side. Chris Fuchs has been an indispensable quantum-foundational critic and consultant. Bob Constable made me, initially against my will, a certified Cornell Information Scientist and introduced me to many members of that excellent community. But most of all, I owe thanks to David DiVincenzo, who collaborated with me on the 1999 two-week LASSP Autumn School and has acted repeatedly over the following years as a sanity check on my ideas, an indispensable source of references and historical information, a patient teacher, and an encouraging friend.

A note on references

Quantum Computer Science is a pedagogical introduction to the basic structure and procedures of the subject – a quantum-computational primer. It is not a historical survey of the development of the field. Many of these procedures are named after the people who first put them forth, but although I use their names, I do not cite the original papers unless they add something to my own exposition. This is because, not surprisingly, work done since the earliest papers has led to clearer expositions of those ideas. I learned the subject myself almost exclusively from secondary, tertiary, or even higher-order sources, and then reformulated it repeatedly in the course of teaching it for six years.

On the few occasions when I do cite a paper it is either because it completes an exposition that I have only sketched, or because the work has not yet become identified in the field with the name(s) of the author(s) and I wanted to make clear that it was not original with me.

Readers interested in hunting down earlier work in the field can begin (and in most cases conclude) their search at the quantum-physics subdivision of the Cornell (formerly Los Alamos) E-print Archive, `http://arxiv.org/archive/quant-ph`, where most of the important papers in the field have been and are still being posted.

Contents

Chapter 1

Cbits and Qbits

1.1 What is a quantum computer?

It is tempting to say that a quantum computer is one whose operation is governed by the laws of quantum mechanics. But since the laws of quantum mechanics govern the behavior of all physical phenomena, this temptation must be resisted. Your laptop operates under the laws of quantum mechanics, but it is not a quantum computer. A quantum computer is one whose operation exploits certain very special transformations of its internal state, whose description is the primary subject of this book. The laws of quantum mechanics allow these peculiar transformations to take place under very carefully controlled conditions.

In a quantum computer the physical systems that encode the individual logical bits must have no physical interactions whatever that are not under the complete control of the program. All other interactions, however irrelevant they might be in an ordinary computer – which we shall call *classical* – introduce potentially catastrophic disruptions into the operation of a quantum computer. Such damaging encounters can include interactions with the external environment, such as air molecules bouncing off the physical systems that represent bits, or the absorption of minute amounts of ambient radiant thermal energy. There can even be disruptive interactions between the computationally relevant features of the physical systems that represent bits and other features of those same systems that are associated with computationally irrelevant aspects of their internal structure. Such destructive interactions, between what matters for the computation and what does not, result in *decoherence*, which is fatal to a quantum computation.

To avoid decoherence individual bits cannot in general be encoded in physical systems of macroscopic size, because such systems (except under very special circumstances) cannot be isolated from their own irrelevant internal properties. Such isolation can be achieved if the bits are encoded in a small number of states of a system of atomic size, where extra internal features do not matter, either because they do not exist, or because they require unavailably high energies to come into play. Such atomic-scale systems must also be decoupled from their surroundings except for the completely controlled interactions that are associated with the computational process itself.

Two things keep the situation from being hopeless. First, because the separation between the discrete energy levels of a system on the atomic scale can be enormously larger than the separation between the levels of a large system, the dynamical isolation of an atomic system is easier to achieve. It can take a substantial kick to knock an atom out of its state of lowest energy. The second reason for hope is the discovery that errors induced by extraneous interactions can actually be corrected if they occur at a sufficiently low rate. While error correction is routine for bits represented by classical systems, quantum error correction is constrained by the formidable requirement that it be done without knowing either the original or the corrupted state of the physical systems that represent the bits. Remarkably, this turns out to be possible.

Although the situation is therefore not hopeless, the practical difficulties in the way of achieving useful quantum computation are enormous. Only a rash person would declare that there will be no useful quantum computers by the year 2050, but only a rash person would predict that there will be. Never mind. Whether or not it will ever become a practical technology, there is a beauty to the theory of quantum computation that gives it a powerful appeal as a lovely branch of mathematics, and as a strange generalization of the paradigm of classical computer science, which had completely escaped the attention of computer scientists until the 1980s. The new paradigm demonstrates that the theory of computation can depend profoundly on the physics of the devices that carry it out. Quantum computation is also a valuable source of examples that illustrate and illuminate, in novel ways, the mysterious phenomena that quantum behavior can give rise to.

For computer scientists the most striking thing about quantum computation is that a quantum computer can be vastly more efficient than anything ever imagined in the classical theory of computational complexity, for certain computational tasks of considerable practical interest. The time it takes the quantum computer to accomplish such tasks scales up much more slowly with the size of the input than it does in any classical computer. Much of this book is devoted to examining the most celebrated examples of this speed-up.

This exposition of quantum computation begins with an introduction to quantum mechanics, specially tailored for this particular application. The quantum-mechanics lessons are designed to give you, as efficiently as possible, the conceptual tools needed to delve into quantum computation. This is done by restating the rules of quantum mechanics, not as the remarkable revision of classical Newtonian mechanics required to account for the behavior of matter at the atomic and subatomic levels, but as a curious generalization of rules describing an ordinary classical digital computer. By focusing exclusively on how quantum mechanics enlarges the possibilities for the physical manipulation of digital information, it is possible to characterize how

the quantum theory works in an elementary and quite concise way, which is nevertheless rigorous and complete for this special area of application.

While I assume no prior familiarity with quantum physics (or any other kind of physics), I do assume familiarity with elementary linear algebra and, in particular, with the theory of finite-dimensional vector spaces over the complex numbers. Appendix A summarizes the relevant linear algebra. It is worth examining even if you are well acquainted with the mathematics of such vector spaces, since it also provides a compact summary of the mathematically unconventional language – *Dirac notation* – in which linear algebra is couched in all treatments of quantum computation. Dirac notation is also developed, more informally, throughout the rest of this chapter.

1.2 Cbits and their states

We begin with an offbeat formulation of what an ordinary classical computer does. I frame the elementary remarks that follow in a language which may look artificial and cumbersome, but is designed to accommodate the richer variety of things that a computer can do if it takes full advantage of the possibilities made available by the quantum-mechanical behavior of its constituent parts. By introducing and applying the unfamiliar nomenclature and notation of quantum mechanics in a familiar classical context, I hope to make a little less strange its subsequent extension to the broader quantum setting.

A classical computer operates on strings of zeros and ones, such as 110010111011000, converting them into other such strings. Each position in such a string is called a *bit*, and it contains either a 0 or a 1. To represent such collections of bits the computer must contain a corresponding collection of physical systems, each of which can exist in two unambiguously distinguishable physical states, associated with the value (0 or 1) of the abstract bit that the physical system represents. Such a physical system could be, for example, a switch that could be open (0) or shut (1), or a magnet whose magnetization could be oriented in two different directions, "up" (0) or "down" (1).

It is a common practice in quantum computer science to use the same term "bit" to describe the two-state classical system that represents the value of the abstract bit. But this use of a single term to characterize both the abstract bit (0 or 1) and the physical system whose two states represent the two values is a potential source of confusion. To avoid such confusion, I shall use the term *Cbit* ("C" for "classical") to describe the two-state classical physical system and *Qbit* to describe its quantum generalization. This terminology is inspired by Paul Dirac's early use of *c-number* and *q-number* to describe classical quantities and their quantum-mechanical generalizations. "Cbit" and

"Qbit" are preferable to "c-bit" and "q-bit" because the terms themselves often appear in hyphenated constructions.

Unfortunately the preposterous spelling *qubit* currently holds sway for the quantum system. The term *qubit* was invented and first used in print by the otherwise admirable Benjamin Schumacher.[1] A brief history of the term can be found in the acknowledgments at the end of his paper. Although "qubit" honors the English (German, Italian, ...) rule that q should be followed by u, it ignores the equally powerful requirement that qu should be followed by a vowel. My guess is that "qubit" has gained acceptance because it visually resembles an obsolete English unit of distance, the homonymic *cubit*. To see its ungainliness with fresh eyes, it suffices to imagine that Dirac had written *qunumber* instead of *q-number*, or that one erased transparencies and cleaned one's ears with *Qutips*.

Because clear distinctions among bits, Cbits, and Qbits are crucial in the introduction to quantum computation that follows, I shall use this currently unfashionable terminology. If you are already addicted to the term *qubit*, please regard *Qbit* as a convenient abbreviation.

To prepare for the extension from Cbits to Qbits, I introduce what may well strike you as a degree of notational overkill in the discussion of Cbits that follows. We shall represent the state of each Cbit as a kind of box, depicted by the symbol $|\ \rangle$, into which we place the value, 0 or 1, represented by that state. Thus the two distinguishable states of a Cbit are represented by the symbols $|0\rangle$ and $|1\rangle$. It is the common practice to call the symbol $|0\rangle$ or $|1\rangle$ itself the *state* of the Cbit, thereby using the same term to refer to both the physical condition of the Cbit and the abstract symbol that represents that physical condition. There is nothing unusual in this. For example one commonly uses the term "position" to refer to the symbol x that represents the physical position of an object. I call this common, if little noted, practice to your attention only because in the quantum case "state" refers *only* to the symbol, there being *no* internal property of the Qbit that the symbol represents. The subtle relation between Qbits and their state symbol will emerge later in this chapter.

Along the same lines, we shall characterize the states of the five Cbits representing 11001, for example, by the symbol

$$|1\rangle|1\rangle|0\rangle|0\rangle|1\rangle, \tag{1.1}$$

and refer to this object as the *state* of all five Cbits. Thus a pair of Cbits can have (or "be in") any of the four possible states

$$|0\rangle|0\rangle,\ |0\rangle|1\rangle,\ |1\rangle|0\rangle,\ |1\rangle|1\rangle, \tag{1.2}$$

1 Benjamin Schumacher, "Quantum coding," *Physical Review* A **51**, 2738–2747 (1995).