Quantum Computer Science
An Introduction

Table of Contents

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 1980's and whose power was not fully appreciated until the mid 1990's.

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, and 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 which 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 some day 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 possible 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 (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 abstract 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 an abstract 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 from CS. 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 may 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 CS 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.