Last revised 1/31/06

LECTURE NOTES ON QUANTUM COMPUTATION

Cornell University, Physics 481-681, CS 483; Spring, 2006

I. Fundamental Properties of Cbits and Qbits

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. The laws of quantum mechanics allow these peculiar transformations under very carefully controlled conditions.

For a computer to be a quantum computer the physical systems that encode the individual 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" when we wish to contrast it to a quantum computer — introduce potentially catastrophic disruptions into the operation of a quantum computer. Such disastrous interactions can include interactions with the external environment — air molecules bouncing off the physical systems that represent bits, or those systems absorbing a minute amount of ambient radiant thermal energy. There can even be disruptive interactions between the computationally relevant degrees of freedom of the physical systems that represent bits with other degrees of freedom of those same systems, associated with computationally irrelevant features of their internal structure. All such interactions between what is computationally relevant and what is not are said to result in "decoherence", which is fatal to a quantum computation.

To avoid decoherence individual bits cannot be encoded in physical systems of macroscopic size, because such systems cannot be isolated from their own irrelevant internal degrees of freedom. The bits must be encoded in a very small number of states of a system of atomic size, where extra internal degrees of freedom do not come into play because they do not exist, or because they require unavailably large amounts of energy to excite. 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 ground state. The second reason for hope is the discovery that errors induced by

extraneous interactions can actually be corrected, provided they occur at a sufficently 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 in the absence of any knowledge of what either the original or the corrupted state of the bits might actually be. Remarkably, this turns out to be possible.

But although the situation is 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 (a) as a lovely branch of abstract mathematics, (b) as a generalization of the paradigm of classical computer science that had completely escaped the attention of computer scientists until the 1980's, demonstrating that in a very deep sense the theory of computation cannot be divorced from the physics of the devices that carry out the computation, and (c) as a source of new examples that illustrate and illuminate the surprising and often mysterious kinds of phenomena that the quantum behavior of matter can give rise to.

You may or may not find point (a) compelling. Many physicists (not including me) are immune to the songs of this particular siren.

The most striking manifestation of point (b) is that for certain special computational tasks of practical interest, a quantum computer can be vastly more efficient than anything ever imagined in the classical theory of computational complexity, in that the time it takes the quantum computer to accomplish the task scales up much more slowly with the size of the input than it can in any classical computer. Much of what follows will be devoted to examining the most celebrated examples of this speed-up.

Item (c) brings us to our first topic. Several years ago I mentioned to a distinguished theoretical physicist (an authority on string theory and director of a great theoretical physics institute, who was later awarded a Nobel prize in physics for his work on quark confinement) that I spent the first four or five 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 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 contribute importantly to the theory of quantum computation. I believe 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 analyzed. Much of the analytical complexity of learning quantum mechanics is connected to 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 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 the real phenomena, 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 the 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 of doing in principle, then there is no reason to get involved in the really difficult physics of the subject.

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 here is extremely focused and quite limited in its scope, you can also be assured that it is neither oversimplified nor incomplete, when applied to the special task for which it is intended.

It might also be noted that another 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. One of the major appeals of quantum computation, at least for me, is that it affords a new conceptual arena for trying to come to a better understanding of quantum weirdness. When opportunities arise I will try to call attention to some of this strange behavior, rather than letting it pass by unremarked upon and unnoticed. One example is described in Section A3 of the appendix to this chapter.

We begin our survey of quantum computation with a minimalist introduction to quantum mechanics, designed to give you, as quickly as possible, the conceptual tools you need to delve into the theories of quantum computation and quantum information processing. I do this by restating the fundamentals 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 way to generalize the behavior of an ordinary classical digital computer. By focusing on the special topic of 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 complete for this restricted area of application.

While I assume no prior familiarity with quantum mechanics, I do assume that you are well acquainted with linear algebra and, in particular, with the theory of (finite-dimensional) vector spaces over the complex numbers.[1]

A. Cbits and their states.

We begin with a minimalist statement of what an ordinary classical computer does. I shall frame the elementary — indeed banal — remarks that follow in a language which, though it may look artificial and cumbersome, is designed to accomodate the richer variety of things that a computer can do if it is designed to take full advantage of the possibilities made available by the quantum mechanical behavior of its constituent parts. Introducing and applying the unfamiliar nomenclature and notation of quantum mechanics in a familiar classical setting should make its subsequent extension to the broader quantum context look a little less peculiar.

A classical computer operates on strings of 0's and 1's, 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 a 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 which 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

[1] For a concise review of linear algebra with subsequent applications to quantum information processing very much in mind, see section 2.1 of *Quantum Computation and Quantum Information*, Michael A. Nielsen and Isaac L. Chuang, Cambridge University Press, 2000. This is an excellent and quite thorough textbook introduction to the whole subject of quantum computation and quantum information. Another fine introduction to the subject can be found in John Preskill's Caltech lecture notes, available at `http://www.theory.caltech.edu/people/preskill/ph229/`.

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.[2] 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 ancient 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 between bits, Cbits, and Qbits are crucial in the introduction to quantum computation that follows, I shall use this unfashionable but superior terminology.

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 rather 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 5 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,

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

three Cbits can be in any of the eight possible states,

$$|0\rangle|0\rangle|0\rangle,\ |0\rangle|0\rangle|1\rangle,\ |0\rangle|1\rangle|0\rangle,\ |0\rangle|1\rangle|1\rangle,\ |1\rangle|0\rangle|0\rangle,\ |1\rangle|0\rangle|1\rangle,\ |1\rangle|1\rangle|0\rangle,\ \text{or}\ |1\rangle|1\rangle|1\rangle, \tag{1.3}$$

[2] The term *qubit* seems first to have been used in print by Benjamin Schumacher, "Quantum Coding", Phys. Rev. A **51**, 2738-2747 (1995). A brief history of the term can be found in the Acknowledgments at the end of Schumacher's paper.

and so on.

As (1.3) already makes evident, when there are many Cbits such products are often much easier to read if one encloses the whole string of 0's and 1's in a single bigger box of the form $|\quad\rangle$ rather than having a separate box for each Cbit:

$$|000\rangle,\ |001\rangle,\ |010\rangle,\ |011\rangle,\ |100\rangle,\ |101\rangle,\ |110\rangle,\ \text{or}\ |111\rangle. \tag{1.4}$$

We shall freely move between these two equivalent ways of expressing the state of several Cbits that represent a string of bits, boxing the whole string, or boxing each individual bit. Whether the form (1.3) or (1.4) is to be preferred depends on the context.

There is also a third form, which is useful when we regard the 0's and 1's as constituting the binary expansion of an integer. We can then replace the representations of the 3-Cbit states (1.4) by the even shorter forms:

$$|0\rangle,\ |1\rangle,\ |2\rangle,\ |3\rangle,\ |4\rangle,\ |5\rangle,\ |6\rangle,\ \text{or}\ |7\rangle. \tag{1.5}$$

Note, though, that unlike the forms (1.3) and (1.4), the form (1.5) is ambiguous, unless we are told that these symbols express the state of 3 Cbits. If we are not told, then there is no way of telling, for example, whether $|3\rangle$ represents the 2-Cbit state $|11\rangle$ or the 3-Cbit state $|011\rangle$ or the 4-Cbit state $|0011\rangle$, etc. This ambiguity can be removed, when necessary, by adding a subscript making the number of Cbits explicit:

$$|0\rangle_3,\ |1\rangle_3,\ |2\rangle_3,\ |3\rangle_3,\ |4\rangle_3,\ |5\rangle_3,\ |6\rangle_3,\ \text{or}\ |7\rangle_3. \tag{1.6}$$

Be warned, however, that when there is no ambiguity about how many Cbits $|x\rangle$ represents, it can be useful to use such subscripts for other purposes. If, for example, Alice and Bob[3] each possess a single Cbit it can be convenient to describe the state of Alice's Cbit (if it has the value 1) by $|1\rangle_a$, Bob's (if it has the value 0) by $|0\rangle_b$, and the joint state of the two by $|1\rangle_a|0\rangle_b$ or $|10\rangle_{ab}$.

Dirac introduced the $|\ \rangle$ notation (known as Dirac notation) in the early days of the quantum theory, as a useful way to write and manipulate *vectors.* For silly reasons — see Section F below — he called such vectors *kets*, a terminology that has survived to this day. In Dirac notation you can put into the box $|\ \rangle$ anything that serves to specify what the vector is. If, for example, we were talking about displacement vectors in ordinary 3-dimensional space, we could have a vector

$$|5\ \text{horizontal centimeters northeast}\rangle. \tag{1.7}$$

In using Dirac notation to express the state of a Cbit, or a collection of Cbits, I'm suggesting that there might be some utility in thinking of the states as vectors. Is there? Well in the case of Cbits, not very much, but maybe a little.

[3] Alice and Bob, long the heroine and hero of diverse cryptographic adventures, play major roles in the exposition of quantum computation and information processing.

We shall briefly explore what one can do with Cbits when one takes the two states $|0\rangle$ and $|1\rangle$ of a single Cbit to be represented by two *orthogonal unit vectors in a 2-dimensional space.* While this is little more than a curious and unnecessarily elaborate way of describing Cbits, it is fundamental and unavoidable in dealing with Qbits. Playing unfamiliar and somewhat silly games with Cbits will, I hope, provide a painless way of becoming introduced to some of the quantum mechanical formalism in a familiar setting. If you prefer your vectors to be expressed in components, note that we can represent the two orthogonal states of a single Cbit, $|0\rangle$ and $|1\rangle$, as column vectors

$$|0\rangle \longleftrightarrow \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad |1\rangle \longleftrightarrow \begin{pmatrix} 0 \\ 1 \end{pmatrix}. \tag{1.8}$$

In the case of two Cbits the vector space is 4-dimensional, with an orthonormal basis

$$|00\rangle,\ |01\rangle,\ |10\rangle,\ |11\rangle. \tag{1.9}$$

The alternative notation for this basis,

$$|0\rangle|0\rangle,\ |0\rangle|1\rangle,\ |1\rangle|0\rangle,\ |1\rangle|1\rangle, \tag{1.10}$$

is deliberately designed to suggest multiplication, since it is, in fact, a short-hand notation for the *tensor products* of the two single-Cbit 2-vectors, written in more formal mathematical notation as[4]

$$|0\rangle \otimes |0\rangle, \quad |0\rangle \otimes |1\rangle, \quad |1\rangle \otimes |0\rangle, \quad |1\rangle \otimes |1\rangle. \tag{1.11}$$

I shall freely move back and forth between these various ways of writing the tensor product, trying in each case to choose the form that makes the content easiest to read.

Once one agrees to regard the two 1-Cbit states as orthogonal unit vectors, the tensor product is indeed the natural way to represent multi-Cbit states, since it leads to the obvious multi-Cbit generalization of the representation (1.8) of 1-Cbit states as column vectors. If we express the states $|0\rangle$ and $|1\rangle$ of each single Cbit as column vectors, then we can get the column-vector describing a multi-Cbit state by applying the standard rule for the components of the tensor product of several two-dimensional vectors, illustrated here

[4] I remind you that the tensor product $\mathbf{a}\otimes\mathbf{b}$ of an M-component vector $\mathbf{a}$ with components a_μ and an N-component vector $\mathbf{b}$ with components b_ν is the (MN)-component vector with components indexed by all the MN possible pairs of indices (μ,ν), whose $(\mu,\nu)^{\text{th}}$ component is just the product $a_\mu b_\nu$.

for a three-fold tensor product:[5]

$$\begin{pmatrix} x_0 \\ x_1 \end{pmatrix} \otimes \begin{pmatrix} y_0 \\ y_1 \end{pmatrix} \otimes \begin{pmatrix} z_0 \\ z_1 \end{pmatrix} \longleftrightarrow \begin{pmatrix} x_0y_0z_0 \\ x_0y_0z_1 \\ x_0y_1z_0 \\ x_0y_1z_1 \\ x_1y_0z_0 \\ x_1y_0z_1 \\ x_1y_1z_0 \\ x_1y_1z_1 \end{pmatrix}. \tag{1.12}$$

Applying this, for example, to the case $|5\rangle_3$ we have

$$|5\rangle_3 = |101\rangle = |1\rangle|0\rangle|1\rangle \longleftrightarrow \begin{pmatrix} 0 \\ 1 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}. \tag{1.13}$$

If we label the vertical components of the 8-vector on the right 0,1,. . . ,7, from the top down, then the single non-zero component is the 1 in the position, 5, specified by the binary number 101 that the three-bit state vector specifies in its original form on the left of (1.13). This is indeed the obvious multi-Cbit generalization of the column-vector form (1.8) for 1-Cbit states.

This is quite general: the tensor product structure of multi-Cbit states is just what one needs in order for the 2^n-dimensional column vector representing a particular one of the 2^n possible states of n Cbits, to have all its entries zero except for a single 1 in the position down from the top specified by the number that those n Cbits are representing in binary.

Indeed, one can turn this development upside down, taking as one's starting point the simple rule that an integer x in the range $0 \leq x < N$ is represented by one of N orthonormal vectors in an N-dimensional space. One can then pick a basis so that 0 is represented by an N-component column vector $|0\rangle$ which has 0 in every position except for a 1 in the top position, and x is to be represented by an N-component column vector $|x\rangle$ which has 0 in every position except for a 1 in the position x down from the top. It then follows from the nature of the tensor product that when $N = 2^n$ and x has the binary expansion $x = \sum_{j=0}^{n-1} x_j 2^j$, then the column vector $|x\rangle$ is the tensor product of the n 2-component column vectors $|x_j\rangle$:

$$|x\rangle = |x_{n-1}\rangle \otimes \cdots \otimes |x_0\rangle. \tag{1.14}$$

[5] This follows from two successive applications of the rule enunciated in the preceding footnote.

This relation between tensor products of vectors and positional notation for integers is not confined to the binary system. If, for example, one represents a decimal digit $x = 0, 1, \ldots, 9$ as a 10-component column vector $\mathbf{v}^{(x)}$ with all components 0 except for a 1, x positions down from the top, then if the n-digit decimal number $X = \sum_{j=0}^{n-1} x_j 10^j$ is represented by the tensor product $\mathbf{V} = \mathbf{v}^{(x_{n-1})} \otimes \mathbf{v}^{(x_{n-2})} \otimes \cdots \otimes \mathbf{v}^{(1)} \otimes \mathbf{v}^{(0)}$, then $\mathbf{V}$ will be a 10^n-component column vector with all components 0 except for a 1, X positions down from the top.

Although the representation of Cbit states by column vectors helps to motivate the use of the tensor product to describe multi-Cbit states, I emphasize that for almost all other purposes it is much better and simpler to forget about column vectors and components, and deal directly with the state vectors in their various abstract forms (1.3)-(1.6).

B. Reversible Operations on Cbits.

Quantum computers do an important part of their magic through *reversible* operations, which transform the initial state of the Qbits into its final form using only processes whose action can be inverted. There is only one *irreversible* part of the operation of a quantum computer. It is called *measurement*, and is the only way to extract useful information from the Qbits after their state has acquired its final form. Although measurement is a nontrivial and crucial part of the quantum computational process, in a classical computer the extraction of information from the state of Cbits is so straightforward a procedure that it is rarely even described as part of the computational process, though it is, of course, a nontrivial concern for those who design digital displays or printers. But all computationally relevant operations in a classical computer correspond to reversible operations on a quantum computer, and therefore only reversible operations on Cbits will be of interest to us here.

In a reversible operation every final state arises from a unique initial state. An example of an irreversible operation is ERASE which forces a Cbit into the state $|0\rangle$ regardless of whether its initial state is $|0\rangle$ or $|1\rangle$. ERASE is irreversible in the sense that given only the final state and the fact that it was the output of the operation ERASE, there is no way to recover the initial state.

The only non-trivial reversible operation we can apply to a single Cbit is the NOT (or *flip*) operation, denoted by the symbol $\mathbf{X}$, which interchanges (or *flips*) the two states $|0\rangle$ and $|1\rangle$:

$$\mathbf{X} : |x\rangle \to |\overline{x}\rangle; \quad \overline{1} = 0, \quad \overline{0} = 1. \tag{1.15}$$

It is reversible because it has an inverse: applying NOT a second time brings the state of the Cbit back to its original form. If we represent the two orthogonal states of the Cbit by the column vectors (1.8), then we can express NOT by a linear operator $\mathbf{X}$ on the 2-dimensional vector space, whose action on the column vectors is given by the matrix

$$\mathbf{X} \longleftrightarrow \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}. \tag{1.16}$$

So the two reversible things you can do to a single bit — leaving it alone or flipping its state — correspond to the two linear operators $\mathbf{X}$ and $\mathbf{1}$,

$$\mathbf{1} \longleftrightarrow \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \tag{1.17}$$

on its two-dimensional vector space.[6]

It can be useful to introduce a *number operator* $\mathbf{n}$ for a single Cbit, defined by

$$\mathbf{n}|x\rangle = x|x\rangle, \quad x = 0 \text{ or } 1; \tag{1.18}$$

i.e. $|0\rangle$ and $|1\rangle$ are eigenvectors of $\mathbf{n}$ with eigenvalues 0 and 1. While multiplying the state of a Cbit by 1 (i.e. leaving it alone) makes sense, you may well ask what it means to multiply its state by 0. (It does not mean removing the Cbit from the computer!) The answer is that we shall never use $\mathbf{n}$ by itself but only in combination with other operations whose combined effect on the states of Cbits does have a straightforward physical meaning.

It is also convenient to define the complementary operator,

$$\overline{\mathbf{n}} = \mathbf{1} - \mathbf{n}, \tag{1.19}$$

so that $|0\rangle$ and $|1\rangle$ are eigenvectors of $\overline{\mathbf{n}}$ with eigenvalues 1 and 0. These operators have the matrix representations

$$\mathbf{n} \longleftrightarrow \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}, \quad \overline{\mathbf{n}} \longleftrightarrow \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}. \tag{1.20}$$

It follows directly from their definitions that

$$\mathbf{n}^2 = \mathbf{n}, \quad \overline{\mathbf{n}}^2 = \overline{\mathbf{n}}, \quad \mathbf{n}\overline{\mathbf{n}} = \overline{\mathbf{n}}\mathbf{n} = \mathbf{0}, \quad \mathbf{n} + \overline{\mathbf{n}} = \mathbf{1}. \tag{1.21}$$

We also have

$$\mathbf{n}\mathbf{X} = \mathbf{X}\overline{\mathbf{n}}, \quad \overline{\mathbf{n}}\mathbf{X} = \mathbf{X}\mathbf{n}, \tag{1.22}$$

since flipping the state of a Cbit and then acting on it with $\mathbf{n}$ ($\overline{\mathbf{n}}$) is the same as acting on the state with $\overline{\mathbf{n}}$ ($\mathbf{n}$) and then flipping it. To complete this collection of trivial identities, note that

$$\mathbf{X}^2 = \mathbf{1}: \tag{1.23}$$

the NOT operator is its own inverse. All the simple relations (1.21)-(1.23) also follow, as they must, from the matrix representations (1.16) and (1.20) for $\mathbf{X}$, $\mathbf{n}$, and $\overline{\mathbf{n}}$.

[6] A pedantic point: Since multiplication by the scalar 1 and action by the unit operator $\mathbf{1}$ achieve the same result, I shall sometimes follow the possibly irritating practice of physicists and not distinguish notationally between them. The same applies to the scalar 0, the zero vector $\mathbf{0}$, and the zero operator $\mathbf{0}$.

The possibilities for reversible operations get richer when we go from a single Cbit to a pair of Cbits. One important operation you can perform on a pair of Cbits is the *swap* (or *exchange*) operation $\mathbf{S}$, which simply interchanges the states of the two. Since $\mathbf{S}$ acts as the identity if the states of the Cbits are the same, and it flips both Cbits if their states are different, it can be written as[7]

$$\mathbf{S} = \mathbf{n} \otimes \mathbf{n} + \overline{\mathbf{n}} \otimes \overline{\mathbf{n}} + (\mathbf{X} \otimes \mathbf{X})(\mathbf{n} \otimes \overline{\mathbf{n}}) + (\mathbf{X} \otimes \mathbf{X})(\overline{\mathbf{n}} \otimes \mathbf{n}). \tag{1.24}$$

Here the tensor product $\otimes$ of two 1-bit operators[8] is the 2-bit operator that acts on the left Cbit with the operator on the left of $\otimes$ and the right Cbit with the operator on the right; i.e.

$$(\mathbf{a} \otimes \mathbf{b})\ |x\rangle \otimes |y\rangle = \mathbf{a}|x\rangle \otimes \mathbf{b}|y\rangle, \tag{1.25}$$

from which it follows that[9]

$$(\mathbf{a} \otimes \mathbf{b})(\mathbf{c} \otimes \mathbf{d}) = (\mathbf{ac}) \otimes (\mathbf{bd}). \tag{1.26}$$

At the risk of belaboring the obvious, I note that (1.24) acts as the swap operator because if both Cbits are in the state $|1\rangle$ (so swapping their states does nothing) then only the first term in the sum acts[10] (and multiplies the state by 1), if both Cbits are in the state $|0\rangle$ only the second term acts (and again multiplies the state by 1), if the left Cbit is in the state $|1\rangle$ and the right Cbit is in the state $|0\rangle$ only the third term acts and the effect of flipping both Cbits is to swap their states, and if the left Cbit is in the state $|0\rangle$ and the right Cbit is in the state $|1\rangle$, only the fourth term acts and the effect of the two $\mathbf{X}$'s is again to swap their states.

The tensor product notation for operators can become quite horrible when one is dealing with a large number of Cbits and wants to write a 2-bit operator that affects only a particular pair of Cbits. If, for example, the 2-bit operator in (1.25) acts only on the

[7] It might be worth making explicit the fact that I am moving back and forth without comment between two levels of description: (1) physical operations on physical Cbits, and (2) linear operators acting on the vectors that represent the states of those Cbits. In doing so I am also blurring the distinction between states (of Cbits) and the vectors that represent those states and between operations (on Cbits) and operators acting on the vectors that represent their states. Indeed, in many contexts "state" and "vector" are taken to be synonymous terms, a practice we shall sometimes follow.

[8] We shall call operations that act jointly on n Cbits n-bit operators, rather than n-Cbit operators, because we shall be extending such operators to act on Qbits as well as Cbits.

[9] While I assume you are familiar with the basic concepts of linear algebra, I shall occasionally (but unsystematically) remind you of some of them, partly because the notation or terminology you are used to may differ from mine.

[10] By this I mean that each of the other three terms gives 0.

second and fourth Cbits from the right in a 6-Cbit state, then the operator on the 6-Cbit state has to be written as

$$\mathbf{1} \otimes \mathbf{1} \otimes \mathbf{a} \otimes \mathbf{1} \otimes \mathbf{b} \otimes \mathbf{1}. \tag{1.27}$$

To avoid such typographical monstrosities we simplify (1.27) to

$$\mathbf{1} \otimes \mathbf{1} \otimes \mathbf{a} \otimes \mathbf{1} \otimes \mathbf{b} \otimes \mathbf{1} = \mathbf{a}_3\mathbf{b}_1 = \mathbf{b}_1\mathbf{a}_3, \tag{1.28}$$

where the subscript indicates which Cbit the 1-bit operator acts on, and it is understood that those Cbits whose subscripts do not appear remain unmodified — i.e. they are acted on by the unit operator. We label each Cbit by the power of 2 it would represent if the Cbits were representing an integer; i.e. the Cbit on the extreme right is labeled 0, the one to its left, 1, etc. Since the order in which $\mathbf{a}$ and $\mathbf{b}$ are written is clearly immaterial if their subscripts specify different Cbits, the order in which one writes them in (1.28) doesn't matter: 1-bit operators that act on different Cbits commute.

So with this convention[11], if the swap operator $\mathbf{S}$ acts on Cbits i and j, we can rewrite (1.24) as

$$\mathbf{S}_{ij} = \mathbf{n}_i\mathbf{n}_j + \overline{\mathbf{n}}_i\overline{\mathbf{n}}_j + (\mathbf{X}_i\mathbf{X}_j)(\mathbf{n}_i\overline{\mathbf{n}}_j + \overline{\mathbf{n}}_i\mathbf{n}_j). \tag{1.29}$$

To help you become more at home with this notation, you are urged to prove as an exercise in algebra, that using only the relations (1.21)-(1.23) and the fact that one-bit operators acting on different Cbits commute, it follows from (1.29) that swapping twice does indeed do nothing: $\mathbf{S}_{ij}^2 = \mathbf{1}$.

As a second illustration of this way of expressing operations on Cbits, consider the *controlled-not* or cNOT operation, $\mathbf{C}_{ij}$, which turns out to be the great workhorse of two-bit operations in quantum computation. If the value represented by the i-th Cbit (the *control bit*) is 0, $\mathbf{C}_{ij}$ leaves the value represented by the j-th Cbit (the *target bit*) unchanged, but if the control bit is 1, $\mathbf{C}_{ij}$ flips the target bit. In either case the control bit is left unchanged. We can summarize this compactly by writing

$$\mathbf{C}_{10}|x\rangle|y\rangle = |x\rangle|y \oplus x\rangle, \tag{1.30}$$

where $\oplus$ denotes addition modulo 2:

$$y \oplus 0 = y, \quad y \oplus 1 = \overline{y} = 1 - y. \tag{1.31}$$

(Computer scientists like to call $x \oplus y$ the "exclusive OR" (or XOR) of x and y.)

[11] Sometimes we deal with 1-bit operators that have subscripts in their names; under such conditions it is more convenient to indicate which Cbit the operator acts on by a superscript, which I shall enclose in parentheses to avoid confusion with an exponent: thus $\mathbf{X}^{(2)}$ represents the 1-bit operator that flips the third Cbit from the right, but $\mathbf{X}^2$ represents the square of the flip operator (i.e. the unit operator) without reference to which Cbit it acts on.

You can build up a swap out of three cNOT operations:

$$\mathbf{S}_{ij} = \mathbf{C}_{ij}\mathbf{C}_{j\,i}\mathbf{C}_{ij}. \tag{1.32}$$

This can easily be verified by repeated applications of (1.30). It can also be demonstrated using a more algebraic approach. Note first that $\mathbf{C}$ can be expressed in terms of $\mathbf{n}$'s and $\mathbf{X}$'s by

$$\mathbf{C}_{ij} = \overline{\mathbf{n}}_i + \mathbf{X}_j\mathbf{n}_i, \tag{1.33}$$

since if the state of Cbit i is $|0\rangle$ only the first term acts which leaves the states of both Cbits unchanged, but if the state of Cbit i is $|1\rangle$ only the second term acts, which leaves the state of Cbit 1 unchanged, while $\mathbf{X}_j$ flips Cbit j. If you substitute expressions of the form (1.33) for each of the three terms in (1.32), then with the help of (1.21)-(1.23) (and the fact that operators with different subscripts commute) you can show that four of the eight terms into which the products expand vanish and the remaining four can be rearranged to give the swap operator in the form (1.29).

A curious symmetry of the cNOT operator is revealed if we define the operator

$$\mathbf{Z} = \overline{\mathbf{n}} - \mathbf{n} \;\longleftrightarrow\; \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}. \tag{1.34}$$

This is a rather peculiar operator, since applied to $|1\rangle$ it produces $-|1\rangle$ and it is not clear what it means to multiply the vector representing the state of a Cbit by -1 (or any number other than 1). Never mind. We shall see that in quantum computation $\mathbf{Z}$ plays as important a role as the NOT operator $\mathbf{X}$, but for now we are only going to use it combined with other operators to produce perfectly sensible transformations on Cbits.

It follows from (1.22) (or from the matrix representations (1.16) and (1.34)) that the NOT operator $\mathbf{X}$ *anticommutes* with $\mathbf{Z}$:

$$\mathbf{ZX} = -\mathbf{XZ}. \tag{1.35}$$

Since $\overline{\mathbf{n}} + \mathbf{n} = \mathbf{1}$ we can use (1.34) to express $\overline{\mathbf{n}}$ and $\mathbf{n}$ in terms of $\mathbf{1}$ and $\mathbf{Z}$:

$$\mathbf{n} = \tfrac{1}{2}(\mathbf{1} - \mathbf{Z}), \quad \overline{\mathbf{n}} = \tfrac{1}{2}(\mathbf{1} + \mathbf{Z}). \tag{1.36}$$

Using this we can then rewrite the cNOT operator (1.33) in terms of $\mathbf{X}$ and $\mathbf{Z}$ operators:

$$\begin{aligned} \mathbf{C}_{ij} &= \tfrac{1}{2}\big(\mathbf{1} + \mathbf{Z}_i\big) + \tfrac{1}{2}\mathbf{X}_j\big(\mathbf{1} - \mathbf{Z}_i\big) \\ &= \tfrac{1}{2}\big(\mathbf{1} + \mathbf{X}_j\big) + \tfrac{1}{2}\mathbf{Z}_i\big(\mathbf{1} - \mathbf{X}_j\big). \end{aligned} \tag{1.37}$$

(The second form follows from the fact that $\mathbf{X}_j$ and $\mathbf{Z}_i$ commute when $i \neq j$.)

Note that if we were to interchange $\mathbf{X}$ and $\mathbf{Z}$ in the second line of (1.37) we would get back the expression directly above it except for the interchange of i and j. So interchanging

the $\mathbf{X}$ and $\mathbf{Z}$ operators has the effect of switching which bit is the control and which is the target, changing $\mathbf{C}_{ij}$ into $\mathbf{C}_{ji}$. An operator that can produce just this effect is the *Hadamard transformation* (also sometimes called the *Walsh-Hadamard transformation.*),

$$\mathbf{H} = \tfrac{1}{\sqrt{2}}(\mathbf{X} + \mathbf{Z}) \;\longleftrightarrow\; \tfrac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}. \tag{1.38}$$

This is another great work-horse of quantum computation. (Physicists should note here an unfortunate clash between the notations of computer science and physics. Quantum physicists invariably use H to denote the Hamiltonian function (in classical mechanics) or Hamiltonian operator (in quantum mechanics). Fortunately Hamiltonian operators, although of crucial importance in the design of quantum computers, play a very limited role in the general theory of quantum computation, being overshadowed by the unitary transformations that they generate. So we can go along with the computer-science notation without getting into serious trouble.)

Since $\mathbf{X}^2 = \mathbf{Z}^2 = \mathbf{1}$ and $\mathbf{XZ} = -\mathbf{ZX}$ one easily shows from the definition (1.38) of $\mathbf{H}$ in terms of $\mathbf{X}$ and $\mathbf{Z}$ that[12]

$$\mathbf{H}^2 = \mathbf{1} \tag{1.39}$$

and that

$$\mathbf{HXH} = \mathbf{Z}, \quad \mathbf{HZH} = \mathbf{X}. \tag{1.40}$$

This shows how $\mathbf{H}$ can be used to interchange the $\mathbf{X}$ and $\mathbf{Z}$ operators in $\mathbf{C}_{ji}$: it follows from (1.40), together with (1.37) and (1.39), that

$$\mathbf{C}_{ji} = \big(\mathbf{H}_i\mathbf{H}_j\big)\mathbf{C}_{ij}\big(\mathbf{H}_i\mathbf{H}_j\big). \tag{1.41}$$

This simple relation can be put to some quite remarkable uses in a quantum computer, as we shall see.

Of course the action of $\mathbf{H}$ on the state of a Cbit that follows from (1.38),

$$\mathbf{H}|0\rangle = \tfrac{1}{\sqrt{2}}(|0\rangle + |1\rangle), \quad \mathbf{H}|1\rangle = \tfrac{1}{\sqrt{2}}(|0\rangle - |1\rangle), \tag{1.42}$$

makes no sense at all. Nevertheless, when combined with other operations, as on the right side of (1.41), the Hadamard operations result in the perfectly sensible operation given on the left side. In a quantum computer the action of $\mathbf{H}$ on 1-Qbit states turns out to be not only meaningful but also easily implemented, and the possibility of interchangng control and target Qbits in the manner shown in (1.41) turns out to have important consequences.

The most general reversible operation on two Cbits is any permutation of their 4 possible states. There are $4! = 24$ such operations. There are $(2^n)!$ distinct reversible operations on n Cbits, given by all possible permutations $\mathbf{P}$ of their 2^n states.

[12] This also follows, of course, from the matrix representation (1.38) of $\mathbf{H}$.

If you would like a matrix for the cNOT operation in the 4-dimensional 2-Cbit subspace note that if the control bit is given by the entry on left then cNOT leaves $|00\rangle = |0\rangle_2$ and $|01\rangle = |1\rangle_2$ fixed and exchanges $|10\rangle = |2\rangle_2$ and $|11\rangle = |3\rangle_2$. Therefore the $4 \otimes 4$ matrix is just

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}. \tag{1.43}$$

If the control bit is shown on the right then $|01\rangle = |1\rangle_2$ and $|11\rangle = |3\rangle_2$ are interchanged and the matrix is

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}. \tag{1.44}$$

By the same token, since the swap operator $\mathbf{S}$ interchanges $|01\rangle = |1\rangle_2$ and $|10\rangle = |2\rangle_2$, leaving $|00\rangle = |0\rangle_2$ and $|11\rangle = |3\rangle_2$ fixed, its matrix is

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \tag{1.45}$$

Note that the construction (1.32) of $\mathbf{S}$ out of cNOT operators follows from (1.43)-(1.45) using matrix multiplication. As a practical matter, however, it is almost always more efficient to establish operator identities by dealing with them directly as operators (as we did above for (1.32) in two different ways), avoiding matrix representations.

Before turning to the quantum generalization of Cbits, let me show you an alternative form for the swap operator (1.29), as a further exercise in treating operations on classical bits as linear operations on vectors. This new form has a rather more elegant structure than (1.29), though you may well regard it as an utterly perverse way to treat Cbits.

If we use (1.36) to reexpress each $\mathbf{n}$ and $\overline{\mathbf{n}}$ appearing in the swap operator (1.29) in terms of $\mathbf{Z}$, we find that

$$\mathbf{S}_{ij} = \tfrac{1}{2}(\mathbf{1} + \mathbf{Z}_i\mathbf{Z}_j) + \tfrac{1}{2}(\mathbf{X}_i\mathbf{X}_j)(\mathbf{1} - \mathbf{Z}_i\mathbf{Z}_j). \tag{1.46}$$

If we define

$$\mathbf{Y} = \mathbf{Z}\mathbf{X} = -\mathbf{X}\mathbf{Z} \longleftrightarrow \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}, \tag{1.47}$$

we get the more compact form

$$\mathbf{S}_{ij} = \tfrac{1}{2}(\mathbf{1} + \mathbf{X}_i\mathbf{X}_j - \mathbf{Y}_i\mathbf{Y}_j + \mathbf{Z}_i\mathbf{Z}_j). \tag{1.48}$$

We can get rid of the irritating minus sign that mars the elegance of (1.48) by replacing $\mathbf{Y}$ with $-i\mathbf{Y}$. (If you wonder what it means to multiply the vector representing a Cbit by $i = \sqrt{-1}$, I merely remark at this point that it is no more or less mysterious than multiplying it by -1. And like -1, the i drops out of the swap operator itself.) We also give $\mathbf{X}$, $\mathbf{Z}$, and $-i\mathbf{Y}$ new names, adopting a more cumbersome notation much beloved by physicists:

$$\sigma_x = \mathbf{X} \longleftrightarrow \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \ \sigma_y = -i\mathbf{Y} \longleftrightarrow \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \ \sigma_z = \mathbf{Z} \longleftrightarrow \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}. \quad (1.49)$$

The swap operator then becomes

$$\mathbf{S}_{ij} = \tfrac{1}{2}(1 + \sigma_x^{(i)}\sigma_x^{(j)} + \sigma_y^{(i)}\sigma_y^{(j)} + \sigma_z^{(i)}\sigma_z^{(j)}). \quad (1.50)$$

(Physicists might be amused at the simplicity of this "computational" derivation of the form of the exchange operator, compared with the conventional quantum mechanical derivation, which invokes the full apparatus of angular momentum theory.)

The pleasing symmetry brought about by introducing a factor of i in the definition of σ_y is not limited to (1.50). In addition the three σ matrices all square to unity,

$$\sigma_x^2 = \sigma_y^2 = \sigma_z^2 = 1, \quad (1.51)$$

they all anticommute in pairs, and the product of any two of them is simply related to the third:

$$\begin{aligned} \sigma_x\sigma_y &= -\sigma_y\sigma_x = i\sigma_z, \\ \sigma_y\sigma_z &= -\sigma_z\sigma_y = i\sigma_x, \\ \sigma_z\sigma_x &= -\sigma_x\sigma_z = i\sigma_y. \end{aligned} \quad (1.52)$$

Note that the three relations (1.52) differ only by cyclic permutations of x, y, and z.

All the relations (1.51) and (1.52) can be summarized in a single compact identity. Let $\mathbf{a}$ and $\mathbf{b}$ be two three-dimensional vectors,

$$\mathbf{a} = (a_x, a_y, a_z), \quad \mathbf{b} = (b_x, b_y, b_z), \quad (1.53)$$

and let σ be a formal vector whose three components are the operators σ_x, σ_y, and σ_z,

$$\sigma = (\sigma_x, \sigma_y, \sigma_z), \quad (1.54)$$

so that, for example by $\mathbf{a} \cdot \sigma$ we mean the ordinary inner product (or "dot product") of the two vectors $\mathbf{a}$ and σ:

$$\mathbf{a} \cdot \sigma = a_x\sigma_x + a_y\sigma_y + a_z\sigma_z. \quad (1.55)$$

Then all the relations (1.51) and (1.52) imply and are implied by the single identity

$$(\mathbf{a}\cdot\sigma)(\mathbf{b}\cdot\sigma) = \mathbf{a}\cdot\mathbf{b} + i(\mathbf{a}\times\mathbf{b})\cdot\sigma, \tag{1.56}$$

where $\mathbf{a}\times\mathbf{b}$ denotes the vector product (or "cross product") of $\mathbf{a}$ and $\mathbf{b}$.

By building the i into the definition of σ_y we have also made it hermitian, like σ_x and σ_z.[13] Together with the unit matrix $\mathbf{1}$, the matrices σ_x, σ_y, and σ_z form a basis for the 4-dimensional algebra of 2-dimensional matrices of complex numbers: any such matrix is a unique linear combination of these four with complex coefficients. Because the four are all hermitian, any 2-dimensional hermitian matrix A of complex numbers must be a *real* linear combination of the four, and therefore of the form[14]

$$A = a_0 + \mathbf{a}\cdot\sigma, \tag{1.57}$$

where a_0 and the components of the vector $\mathbf{a}$ are all real numbers.

The matrices σ_x, σ_y, and σ_z were introduced in the early days of quantum mechanics by Wolfgang Pauli, to describe the angular momentum associated with the spin of an electron. They have many other useful purposes, being simply related to the quaternions invented by Hamilton to deal efficiently with the composition of three-dimensional rotations.[15] It is pleasing (if somewhat perverse) to find them here, buried in the interior of the operator that simply swaps two classical bits. We shall have extensive occasion to use Pauli's one-bit operators when we come to the subject of quantum error correction, and some of their properties, developed further in sections A2 and A3 of the appendix to this chapter, can be useful in treating other features of Qbits, to which we now turn.

C. Qbits and their states.

The state of a Cbit is a pretty miserable specimen of a two-dimensional vector, since the only vectors with any classical meaning in the whole two-dimensional vector space are the two orthonormal vectors $|0\rangle$ and $|1\rangle$. Putting it another way, each Cbit can have only two states: $|0\rangle$ and $|1\rangle$. Happily, nature has provided us with physical systems, Qbits, described by states that do not suffer from this limitation. The state $|\psi\rangle$ associated with a Qbit can be any unit vector in the two-dimensional space spanned by $|0\rangle$ and $|1\rangle$. Since we have already seen a hint of the elegance the use of $i = \sqrt{-1}$ can introduce even into the

[13] Recall that the elements of a hermitian matrix A satisfy $A_{ji} = A^*_{ij}$, where * denotes complex conjugation.

[14] Unless it is important to emphasize the operator structure, when a_0 is an scalar we shall write expressions such as $a_0\mathbf{1}$ as simply a_0.

[15] Hamilton's quaternions i, j, k are represented by $i\sigma_x, i\sigma_y, i\sigma_z$. The beautiful connection between the Pauli matrices and three-dimensional rotations discovered by Hamilton is developed in Section A2 of the appendix to this chapter.

tightly constrained world of Cbits, you will, I hope, be pleased to learn that the scalars in the two-dimensional vector space containing the states of a Qbit are complex numbers. The general state of a Qbit is

$$|\psi\rangle = \alpha_0|0\rangle + \alpha_1|1\rangle \;\longleftrightarrow\; \begin{pmatrix} \alpha_0 \\ \alpha_1 \end{pmatrix}, \tag{1.58}$$

where α_0 and α_1 are two complex numbers constrained only by the requirement that $|\psi\rangle$, like $|0\rangle$ and $|1\rangle$, should be a unit vector in the complex vector space — i.e. only by the normalization condition

$$|\alpha_0|^2 + |\alpha_1|^2 = 1. \tag{1.59}$$

One says that the state $|\psi\rangle$ is a *superposition* of the states $|0\rangle$ and $|1\rangle$ with *amplitudes* α_0 and α_1.

If one of α_0 and α_1 is 0 and the other is 1 — i.e. in the special case where the state of the Qbit is one of the two classical states $|0\rangle$ or $|1\rangle$ — it is often convenient to retain the language appropriate to Cbits, speaking of the Qbit "having the value" 0 or 1. More correctly, however, one is only entitled to say that the state of the Qbit is $|0\rangle$ or $|1\rangle$. Qbits, in contrast to Cbits, cannot be said to "have values". They only have — or, more correctly, *are described by* — states. We shall often sacrifice correctness for ease of expression. Some reasons for this apparently pedantic terminological hair splitting will emerge below.

Just as the general state of a single Qbit is an arbitrary normalized superposition (1.58) of the two possible classical states, the general state $|\Psi\rangle$ that nature allows us to associate with two Qbits is an arbitrary normalized superposition of the four orthogonal classical states,

$$|\Psi\rangle = \alpha_{00}|00\rangle + \alpha_{01}|01\rangle + \alpha_{10}|10\rangle + \alpha_{11}|11\rangle \;\longleftrightarrow\; \begin{pmatrix} \alpha_{00} \\ \alpha_{01} \\ \alpha_{10} \\ \alpha_{11} \end{pmatrix}, \tag{1.60}$$

with the complex amplitudes being constrained only by the normalization condition

$$|\alpha_{00}|^2 + |\alpha_{01}|^2 + |\alpha_{10}|^2 + |\alpha_{11}|^2 = 1. \tag{1.61}$$

This generalizes in the obvious way to n Qbits, whose general state can be a superposition of the 2^n different classical states, with amplitudes whose squared magnitudes sum to unity:

$$|\Psi\rangle = \sum_{0 \le x < 2^n} \alpha_x |x\rangle_n, \tag{1.62}$$

$$\sum_{0 \le x < 2^n} |\alpha_x|^2 = 1. \tag{1.63}$$

In the context of quantum computation, the 2^n classical states — the set of all possible products of individual Qbit states $|0\rangle$ and $|1\rangle$ — is called the *computational basis.* For many purposes *classical basis* would be a more appropriate term, and I shall use the two terms interchangeably. The states that characterize n Cbits — the classical-basis states — are an extremely limited subset of the states of n Qbits, which can be any (normalized) superposition with complex coefficients of these classical-basis states.

If we have two Qbits, one in the state $|\psi\rangle = \alpha_0|0\rangle + \alpha_1|1\rangle$ and the other in the state $|\phi\rangle = \beta_0|0\rangle + \beta_1|1\rangle$, then the state of the pair, in straightforward generalization of the rule for multi-Cbit states, is taken to be the tensor product of the individual states,

$$\begin{aligned}|\Psi\rangle = |\psi\rangle \otimes |\phi\rangle &= \big(\alpha_0|0\rangle + \alpha_1|1\rangle\big) \otimes \big(\beta_0|0\rangle + \beta_1|1\rangle\big) \\ &= \alpha_0\beta_0|00\rangle + \alpha_0\beta_1|01\rangle + \alpha_1\beta_0|10\rangle + \alpha_1\beta_1|11\rangle \;\longleftrightarrow\; \begin{pmatrix} \alpha_0\beta_0 \\ \alpha_0\beta_1 \\ \alpha_1\beta_0 \\ \alpha_1\beta_1 \end{pmatrix}. \end{aligned} \tag{1.64}$$

Note that a *general* 2-Qbit state (1.60) is of the special form (1.64) if and only if $\alpha_{00}\alpha_{11} = \alpha_{01}\alpha_{10}$. Since the four amplitudes in (1.60) are constrained only by the normalization condition (1.61), this relation need not hold, and the general 2-Qbit state, unlike the general state of two Cbits, is *not* a product (1.64) of two 1-Qbit states. The same is true for states of n Qbits: unlike Cbits, whose general state can only be one of the 2^n products of $|0\rangle$'s and $|1\rangle$'s, a general state of n Qbits is a superposition of these 2^n product states which will not, in general, be any product of any set of 1-Qbit states. Individual Qbits making up a multi-Qbit system, in contrast to individual Cbits, cannot always be characterized as having individual states of their own.[16]

Such nonproduct states of two or more Qbits are called *entangled* states.[17] When the state of several Qbits is entangled, they can sometimes behave in some very strange ways. An example of such peculiar behavior is discussed in Section A3 of the appendix to this chapter.

D. Reversible Operations on Qbits.

The only nontrivial reversible operation a classical computer can perform on a single Cbit is the NOT operation **X**. Nature has been far more versatile in what it allows us to do

[16] More precisely they do not always have what are called *pure states* of their own. It is often convenient to give a statistical characterization of an individual Qbit in terms of what is called a *density matrix* or *mixed state.* If one wishes to emphasize that one is not talking about a mixed state, one uses the term "pure state". In this exposition of quantum computation the term "state" can always be taken to mean "pure state" unless explicitly noted otherwise.

[17] The term is a translation of Schrödinger's *verschränkt*, which I am told is rendered more accurately as "entwined" or "enfolded". But Schrödinger himself used the translation "entangled", and may even have used it first in English, before coining the German term.

to a Qbit. The reversible operations that a quantum computer can perform upon a single Qbit are represented by the action on the state of the Qbit of any *linear* transformation that takes unit vectors into unit vectors. Such transformations $\mathbf{u}$, of which $\mathbf{X}$ is a special case, are called *unitary* and satisfy the condition[18]

$$\mathbf{u}\mathbf{u}^\dagger = \mathbf{u}^\dagger\mathbf{u} = 1. \tag{1.65}$$

Since any unitary transformation has a unitary inverse, such actions of a quantum computer on a Qbit are reversible. The importance of reversibility for the effective functioning of a quantum computer will emerge in Chapter II.

Such unitary operations on a Qbit are called 1-Qbit gates. Their action is represented schematically as in Figure 1.1.

The most general reversible n-Cbit operation in a classical computer is a permutation of the $(2^n)!$ different classical-basis states. The most general reversible operation that a quantum computer can perform upon n-Qbits is represented by the action on the state of those Qbits of any linear transformation that takes unit vectors into unit vectors — i.e. any 2^n-dimensional unitary transformation $\mathbf{U}$, satisfying

$$\mathbf{U}\mathbf{U}^\dagger = \mathbf{U}^\dagger\mathbf{U} = 1. \tag{1.66}$$

Such transformations are called n-Qbit gates.

The action of an n-Qbit unitary gate on n Qbits is represented schematically by the kind of diagram shown in Figure 1.2. Figure 1.3 shows the schematic representation of two successive unitary transformations: first $\mathbf{V}$ is applied to the Qbits, and then $\mathbf{U}$. It is the practice of computer scientists to read such circuit diagrams from left to right, like ordinary prose in European languages, so the the operator $\mathbf{V}$ occurs to the left of $\mathbf{U}$ because it acts on the Qbits before $\mathbf{U}$ acts. The final state, however, is conventionally written as $\mathbf{U}\big(\mathbf{V}|\Psi\rangle\big) = \mathbf{U}\mathbf{V}|\Psi\rangle$, it being the practice of physicsts to write the operators to the left of the state symbol. As an unfortunate result of this intermingling of cultures, the left-to-right order in which operators appear in circuit diagrams is reversed from the left-to-right order in which they appear in the mathematical expression for the state that diagram gives rise to.

Any operation $\mathbf{P}$ that acts on n Cbits, can be associated with a corresponding unitary operation $\mathbf{U}$ on n Qbits. One defines the action of $\mathbf{U}$ on the classical-basis states to be identical to the operation of $\mathbf{P}$ on the corresponding classical states. Since the classical basis *is* a basis, $\mathbf{U}$ can then be extended to arbitrary n-Qbit states by linearity. But since the action of $\mathbf{U}$ on the classical-basis states is merely to permute them, its effect on any superposition of such states is merely to permute the amplitudes. So it trivially takes unit vectors (for which the sums of the moduli of the squared amplitudes is unity) into

[18] Recall that the adjoint of a matrix is the transposed complex conjugate: $(\mathbf{A}^\dagger)_{ij} = (\mathbf{A}_{ji})^*$.

unit vectors for arbitrary quantum states. Being norm-preserving and linear, it is indeed unitary. Many important unitary operations on Qbits we shall be examining below are defined in this way, as linear extensions to Qbits of classical operations on Cbits. But the available unitary transformations on Qbits are, of course, much more general than straightforward extensions of classical operations.

In the actual design of quantum algorithms the class of allowed unitary transformations is almost always restricted to ones that can be built up out of products of unitary transformations that act on only one Qbit at a time (*1-Qbit gates*) or a pair of Qbits (*2-Qbit gates*). This is because the technical problems of making higher order quantum gates are even more formidable than the (already difficult) problems of constructing reliable 1- and 2-Qbit gates. It turns out that this is not a fundamental limitation, since arbitrary unitary transformations can be approximated to an arbitrary degree of precision by sufficiently many 1- and 2-Qbit gates. We shall not prove this general result, because all of the quantum algorithms we shall be developing will be explicitly built up entirely from 1- and 2-Qbit gates. One very important illustration of the sufficiency of 1- and 2-Qbit gates will emerge in Chapter 2: In a reversible classical computer it turns out that one needs 3-Cbit gates to build up general logical operations. But in a quantum computer we shall see that the quantum extensions of such 3-Cbit gates can be constructed out of a small number of 1- and 2-Qbit gates.

While unitarity is generally taken to be the hallmark of the transformations nature allows us to perform on quantum states, it is worth emphasizing that what is really remarkable about these transformations is their *linearity*, which is, of course, one aspect of their unitarity. It is, for example, easy to dream up straightforward classical models for a Qbit, particularly if one restricts its superpositions to real linear combinations of the two classical states. And it is not hard to invent classical models for NOT and even Hadamard 1-Qbit gates, for example, that act linearly on all the 1-Qbit states, while acting as NOT or Hadamard on the classical basis states. But it is not at all easy to dream up a straightforward classical model for a cNOT gate, for example, that acts *linearly* on all the extended 2-Qbit states, while acting as cNOT on the four computational basis states. It is remarkable and highly nontrivial that nature allows us, with much ingenuity and hard work, to fabricate such gates,

E. The measurement of Qbits.

To specify the state of a single Cbit you need only one bit of information: whether the state of the Cbit is $|0\rangle$ or $|1\rangle$. But to specify the state (1.58) of a single Qbit to an arbitrarily high degree of precision you have to specify arbitrarily many bits of information, since you must specify two complex numbers α and β subject only to the normalization constraint (1.59). Because Qbits not only have a much richer set of states than Cbits, but also can be acted on by a correspondingly richer set of transformations, it might appear obvious that a quantum computer would be vastly more powerful than a classical computer. *But there is a major catch!*

The catch is this: if you have n Cbits, each represents either 0 or 1, and you can find out the state of each just by looking. There is nothing problematic about learning the state of a Cbit, and hence learning the result of any calculation you may have built up out of operations on those Cbits. Furthermore — and this is taken completely for granted in any discussion of a classical computer — the state of Cbits is not altered by the process of reading them. The act of acquiring the information from Cbits is not disruptive. You can read the Cbits at any stage of a computation without disrupting subsequent stages.

In stark contrast, if you have n Qbits in a superposition (1.62) of computational basis states, there is nothing whatever you can do to to them to extract from those Qbits the vast amount of information contained in the amplitudes α_x. You cannot read out those values and therefore you cannot find out what the state is. The state of n Qbits is not associated with any ascertainable property of those Qbits, as it is for Cbits. There is only one way to extract information from n Qbits in a given state: it is called *making a measurement*.[19] Making a measurement consists of performing a certain test on each Qbit, the outcome of which is either 0 or 1. The particular collection of 0's and 1's produced by the test is not in general determined by the state $|\Psi\rangle$ of the Qbits; the state determines only the *probability* of the possible outcomes, according to the following rule:

The probability of getting a particular result — say 01100, if you have 5 Qbits — is given by the squared magnitude of the amplitude of the state $|01100\rangle$ in the expansion of the state $|\Psi\rangle$ of the Qbits in the 2^5 computational basis states. More generally, if the state of n Qbits is

$$|\Psi\rangle_n = \sum_{0\le x<2^n} \alpha_x |x\rangle_n, \tag{1.67}$$

then the probabiilty that the 0's and 1's resulting from measurements of all the Qbits will give the binary expansion of the integer x is

$$p(x) = |\alpha_x|^2. \tag{1.68}$$

This basic rule for how information can be extracted from a quantum state was first enunciated by Max Born, and is known as the *Born rule.* It provides the link between amplitudes and the numbers you can actually read out when you test — i.e. measure —

[19] Physicists will note — others need pay no attention to this remark — that what follows is more accurately characterized as "making a (von Neumann) measurement in the computational (classical) basis." There are other ways to make such a measurement, but they can all be reduced to measurements in the computational basis provided an appropriate unitary transformation is applied to the n-Qbit state of the computer just before carrying out the measurement. So when I use the term "measurement", I shall always mean measurement in the computational basis. Measurements in other bases will always be treated as measurements in the computational basis preceded by suitable unitary transformations.

the Qbits. The squared magnitudes of the amplitudes give the probabilities of outcomes of measurements. Normalization conditions like (1.61) are just the requirements that the probabilities for all of the 2^n mutually exclusive outcomes add up to 1.

This process of measurement is carried out by a piece of hardware with a digital display known as an n-Qbit *measurement gate*. Such an n-Qbit "measurement gate" is depicted schematically in Figure 1.4. In contrast to unitary gates, which have a unique output state for each input state, the state of the Qbits emerging from a measurement gate is only *statistically* determined by the state of the input Qbits. In further contrast to unitary gates, the action of a measurement gate cannot be undone: given the final state $|x\rangle$ there is no way of reconstructing the initial state $|\Psi\rangle$. Nor is the action of a measurement gate in any sense linear.

To the extent that it suggests that some preexisting property is being revealed, "measurement" is a highly misleading term, but it is hallowed by three quarters of a century of use by quantum physicists, and impossible to avoid in treatments of quantum computation. One should avoid being misled by such spurious connotations of "measurement" (though it confused many physicists in the early days of quantum mechanics). In quantum computation "measurement" means nothing more or less than applying and reading the display of an appropriate measurement gate, whose action is fully specified by the Born rule, as described above (and expanded on below).[20]

The simplest statement of the Born rule is for a single Qbit. If the state of the Qbit is the superposition (1.58) of the states $|0\rangle$ and $|1\rangle$ with amplitudes α_0 and α_1 then the result of the measurement is 0 with probability $|\alpha_0|^2$ and 1 with probability $|\alpha_1|^2$. The measurement is carried out by a 1-Qbit measurement gate, as illustrated in Figure 1.5.

We shall see below that n-Qbit measurement gates can be realized by applying 1-Qbit measurement gates to each of the n Qbits. The process of measurement can thus be reduced to applications of multiple copies of a single elementary piece of hardware: the 1-Qbit measurement gate.

In addition to displaying an n-bit integer with probabilities determined by the amplitudes, there is a second very important aspect of the action of measurement gates: If n Qbits, initially described by a state $|\Psi\rangle$, are sent through an n-Qbit measurement gate, and the display of the measurement gate indicates the integer x, then one must associate with the Qbits emerging from that measurement gate the classical-basis state $|x\rangle_n$ (as already shown in Figures 1.4 and 1.5). Among other things, this means that all traces of the amplitudes α_x characterizing the input state have vanished from the output state. The only role they have played in the measurement is to determine the probability of a particular output.

[20] While measurement in quantum mechanics is not at all like measuring somebody's weight, it is a bit more like measuring somebody's IQ, which (I would say) reveals no pre-existing property of people, but only what happens when they are subject to the IQ test.

If the state of the input Qbits is one of the classical-basis states $|x\rangle_n$, then according to the Born rule the probability is 1 that the measurement gate will read x and the output state will remain $|x\rangle_n$. But for superpositions (1.67) with more than a single non-zero amplitude α_x, the output state is not determined, and, being a single one of the classical basis states $|x\rangle_n$, no longer carries any information about the amplitudes characterizing the initial state, other than certifying that the particular amplitude α_x was not zero, and, in all likelihood, was not exceedingly small.

So once you send n Qbits through an n-Qbit measurement gate, you remove the possibility of extracting any further information about their original state $|\Psi\rangle$. After such a measurement of 5 Qbits, if the result is 01100, then the post-measurement state associated with the Qbits is no longer $|\Psi\rangle$, but $|01100\rangle$. The original state $|\Psi\rangle$, and all the rich information potentially available in its amplitudes is irretrievably lost. Qbits emerging from a measurement gate that indicates the outcome x are characterized by the state $|x\rangle$, regardless of what their pre-measurement state may have been.

This change of state attendant upon a measurement is often referred to as a *reduction* or *collapse* of the state. One says that as a consequence of the measurement the pre-measurement state *reduces* or *collapses* to the post-measurement state. This should not be taken to imply (though, alas, it often is) that the Qbits themselves suffer a catastrophic "reduction" or "collapse". It is important to keep in mind, in this context, that the state of n Qbits is nothing more than an abstract symbol, used, via the Born rule, to calculate probabilities of measurement outcomes. As noted above, there is no internal property of the Qbits that corresponds to their state.

You might well wonder how one can learn anything at all of computational interest under these wretched conditions. The artistry of quantum computation consists in producing, through a cunningly constructed unitary transformation, a superposition in which most of the amplitudes α_x are zero or extremely close to zero, with useful information being carried by *any* of the values of x that have an appreciable probability of being indicated by the measurement. It is thus important to be seeking information which, once possessed, can easily be confirmed (e.g. the factors of a large number) so that one is not misled by rare and irrelevant low probability outcomes. How this is actually accomplished in various cases of interest will be one of our major preoccupations.

There is a stronger version of the Born rule, which plays a very important role in quantum computation, even though it is rarely explicitly mentioned in most standard quantum mechanics texts. We shall call it the *generalized Born rule.* This stronger form applies when one measures only a single one of n Qbits, using a standard 1-Qbit measurement gate. The generalized Born rule is based on the fact that any state $|\Psi\rangle$ of all n Qbits can be represented in the form

$$|\Psi\rangle = a_0|0\rangle|\Phi_0\rangle + a_1|1\rangle|\Phi_1\rangle, \qquad |a_0|^2 + |a_1|^2 = 1, \tag{1.69}$$

where the Qbit to be measured appears on the left, and where $|\Phi_0\rangle$ and $|\Phi_1\rangle$ are normalized (but not necessarily orthogonal) states of the $n-1$ unmeasured Qbits. This follows

straightforwardly from the fact that the general form (1.67) of $|\Psi\rangle$ is indeed of the form (1.69) with $|\Phi_0\rangle$ and $|\Phi_1\rangle$ given, to within normalization constants, by

$$|\Phi_0\rangle \propto \sum_{x=0}^{2^{n-1}-1} \alpha_x |x\rangle_{n-1}, \quad |\Phi_1\rangle \propto \sum_{x=0}^{2^{n-1}-1} \alpha_{x+2^{n-1}} |x\rangle_{n-1}. \tag{1.70}$$

The generalized Born rule asserts that if only the single Qbit associated with the one-Qbit state on the left of (1.69) is measured, then the 1-Qbit measurement gate will indicate x (0 or 1) with probability $|a_x|^2$, after which the n-Qbit state will be the product state $|x\rangle|\Phi_x\rangle$.

This action of a 1-Qbit measurement gate on an n-Qbit state is depicted schematically in Figure 1.6. Figure 1.7 (and its caption) demonstrates that if the Qbit on which the 1-Qbit gate acts is initially disentangled from the remaining $n-1$ Qbits, then the action of the gate on the measured Qbit is just that specified by the ordinary Born rule, and the unmeasured Qbits play no role at all, remaining throughout the process in their original state. This is also evident from the preceding paragraph, if one notes that it corresponds to the case in which the two not necessarily orthogonal states $|\Phi\rangle_0$ and $|\Phi\rangle_1$ are identical.

If one applies the generalized Born rule n times to successive one-Qbit measurements of each of n Qbits, initially in the general n-Qbit state (1.67), one can show (by a straightforward but irritatingly awkward argument) that the final state of the n-Qbits is x with probability $|\alpha_x|^2$, where x is the n-bit integer whose bits are given by the readings on the n one-Qbit measurement gates. This is nothing but the ordinary Born rule, with the n 1-Qbit measurement gates playing the role of the single n-Qbit measurement gate. There is thus, as remarked upon above, only one primitive piece of measurement hardware: the 1-Qbit measurement gate. This construction of an n-Qbit measurement gate out of n 1-Qbit measurement gates is depicted schematically in Figure 1.8.

An even more general version of the Born rule follows from the generalized Born rule as stated above. The general state of $m+n$ Qbits can be written as

$$|\Psi\rangle_{m+n} = \sum_x \alpha_x |x\rangle_m |\Phi_x\rangle_n \tag{1.71}$$

where $\sum_x |\alpha_x|^2 = 1$ and the states $|\Phi_x\rangle_n$ are normalized, but not necessarily orthogonal. By applying the generalized Born rule m times to m Qbits in an $m+n$ Qbit state, one establishes the the rule that if only the m Qbits on the left of (1.71) are measured, then with probability $|\alpha_x|^2$ the result will be x, and after the measurement the state of all $m+n$ Qbits, will be the product state

$$|x\rangle_m |\Phi_x\rangle_n \tag{1.72}$$

in which the m measured Qbits are in the state $|x\rangle_m$ and the n unmeasured ones are in the state $|\Phi_x\rangle_n$.

It is important to note and immediately reject a possible misunderstanding of the Born rule. One might be tempted to infer from that rule that for a Qbit to be in a

superposition like the state $|\psi\rangle = \alpha_0|0\rangle + \alpha_1|1\rangle$ means nothing more than that the "actual state" of the Qbit is either $|0\rangle$ with probability $|\alpha_0|^2$ or $|1\rangle$ with probability $|\alpha_1|^2$. Such an assertion goes beyond the rule, of course, which merely asserts that if one subjects a Qbit in the state $|\psi\rangle$ to an appropriate test — a measurement — then the outcome of the test will be 0 or 1 with those probabilities and the post-measurement state of the Qbit can correspondingly be taken to be $|0\rangle$ or $|1\rangle$. This does not imply that prior to the test the Qbit already had the value — i.e. already was described by the classical-basis state — revealed by the test, since, among other possibilities, the action of the test itself might well play a role in bringing forth the outcome.

In fact it is easy to produce examples that demonstrate that the Qbit, prior to the test, *could not* have been in either one or the other of the states $|0\rangle$ or $|1\rangle$. We can see this with the help of the Hadamard transformation (1.38). We have defined the action of the 1-Qbit operators $\mathbf{H}$, $\mathbf{X}$, and $\mathbf{Z}$ only on the computational basis states $|0\rangle$ and $|1\rangle$, but we can extend their action to arbitrary linear combination of these states by requiring the extensions to be linear operators. Since the states $|0\rangle$ and $|1\rangle$ form a basis, this determines the action of $\mathbf{H}$, $\mathbf{X}$, and $\mathbf{Z}$ on any 1-Qbit state.

Because it is hermitian and its own inverse, $\mathbf{H}$ is unitary, and is therefore the kind of operation a quantum computer can apply to the state of a Qbit: a 1-Qbit gate. The result of the operation of a Hadamard gate is to change the state $|\phi\rangle$ of a Qbit to $\mathbf{H}|\phi\rangle$. Suppose, now, that we apply $\mathbf{H}$ to a Qbit that is initially in the state

$$|\phi\rangle = \tfrac{1}{\sqrt{2}}(|0\rangle + |1\rangle). \tag{1.73}$$

It follows from (1.42) that the result is just

$$\mathbf{H}|\phi\rangle = |0\rangle. \tag{1.74}$$

So according to the Born rule if we measure a Qbit described by the state $\mathbf{H}|\phi\rangle$ the result will be 0 with probability 1.

But suppose that a Qbit in the state $|\phi\rangle$ were indeed either in the state $|0\rangle$ with probability $\frac{1}{2}$ or in the state $|1\rangle$ with probability $\frac{1}{2}$. In either case, according to (1.42), the subsequent action of $\mathbf{H}$ would produce a state — either $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ or $\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$ — that under measurement yielded 0 or 1 with equal probability. This contradicts the fact just extracted directly from (1.74) that the result of making a measurement on a Qbit in the state $\mathbf{H}|\phi\rangle$ is invariably 0.

So a Qbit in a quantum superposition of $|0\rangle$ and $|1\rangle$ cannot be viewed as being either in the state $|0\rangle$ or in the state $|1\rangle$ with certain probabilities. Such a state represents something quite different from what either of these possibilities represent. Although the Qbit only reveals a 0 or a 1 when you query it by means of a measurement, prior to such a query its state is not in general either $|0\rangle$ or $|1\rangle$, but a superposition of the form (1.58). Such a superposition is as natural and irreducible a description of a Qbit as $|0\rangle$ and $|1\rangle$ are. (See Section A2 of the appendix to this chapter for a further development of this point.)

If the state (1.67) is one of the 2^n computational-basis states $|x\rangle_n$ so that $\alpha_x = 1$ and $\alpha_y = 0$ when $y \neq x$, then the Born rule (1.68) reduces to the assertion that the outcome of the measurement is x with probability 1, and the post-measurement state is $|x\rangle$ — i.e. it is unchanged from the pre-measurement state. So if the states of n Qbits are restricted to computational basis states then the process of measurement is just like the classical process of "learning the value" of x without altering the state. A quantum computer can be made to simulate a reversible classical computer by allowing only computational basis states as input, and only allowing unitary transformations that take computational basis states into computational basis states.

In addition to providing an output at the end of a computation, measurement gates also play a crucial role (not often sufficiently emphasized) at the beginning. If there is no way to determine the state of a given collection of Qbits — indeed, in general such a collection might be entangled with other Qbits and therefore not even have a state of its own — then how can one produce a set of Qbits in a definite state for the gates of a quantum computer to transform into another computationally useful state?

The answer is by measurement. If one takes n Qbits off the shelf, and subjects them to a measurement gate that registers x, then one can be sure that the Qbits emerging from that gate are in the classical-basis state $|x\rangle_n$. If one then applies the 1-Qbit operation $\mathbf{X}$ to each Qbit that registered a 1 in the measurement, doing nothing to the Qbits that registered 0, the resulting set of Qbits will be described by the state $|0\rangle_n$, and it is this state that most quantum-computational algorithms take as their input. Such a use of a measurement gate to provide a Qbit described by the state $|0\rangle$ is shown in Figure 1.9.

Measurement gates therefore play *two* roles in a quantum computation. They get the Qbits ready for the subsequent action of the computer, and after the computer has acted they extract from the Qbits a digital output. The initial action of the measurement gates is often called "state preparation," since the Qbits emerging from the process can be characterized by a definite state. The association of unitary operators with the gates that subsequently act on the Qbits, permits one to update that state assignment into the corresponding unitary transformation of the initial state, thereby making it possible to calculate, using the Born rule, the probabilities of the outcomes of the final measurement gates.

Table: Cbits vs. Qbits

I conclude this offbeat introduction to quantum mechanics with a table that summarizes the relevant physical features of Qbits by contrasting them to the analogous features of Cbits. In the table I introduce the term "Bit", with an upper-case B, to mean "Qbit or Cbit" (as opposed to "bit", with a lower-case b, which means "0 or 1"). Alice (5th line) is anybody who knows the relevant history of the Qbits: the initial "state preparation" and the unitary gates that have subsequently acted.

CLASSICAL vs. QUANTUM BITS	**Cbits**	**Qbits**
States of n Bits	$\|x\rangle_n, \quad 0 \le x < 2^n$	$\sum \alpha_x \|x\rangle_n, \quad \sum \|\alpha_x\|^2 = 1$
Subsets of n Bits	Always have states	Generally have no states
Reversible operations on states	Permutations	Unitary transformations
Can state be learnt from Bits?	Yes	No
To learn state of Bits	Examine them	Go ask Alice
To get information from Bits	Just look at them	Measure them
Information acquired	x	x with probability $\|\alpha_x\|^2$
State after information acquired	Same: still $\|x\rangle$	Different: now $\|x\rangle$

Appendix to Chapter 1

Sections A1-A3 of this appendix give some further mathematical developments of a slightly more technical nature, that we will make only occasional use of. I recommend glancing through them now and examining them more closely when their contents are needed. Section A4 (which is independent of A1-A3) applies some of the quantum formalism developed in the chapter to illustrate a famous and striking peculiarity of Qbits: "Spooky Action at a Distance." Although the subject is of broad cultural interest, it is not directly relevant to quantum computation or quantum information processing. I have included it primarily to give you some further illustrations of how to use the quantum formalism developed in the chapter.

A1. Further Features of Dirac Notation.

The features of Dirac notation developed above are the only ones needed to understand the manipulation of and extraction of information out of Qbits, and to read much of what follows. There are, however, several mathematical embellishments of the notation which can be useful, which you will encounter if you delve into the literature on quantum computation and quantum information, and which come up occasionally in these lecture notes.

There is an inner product between two n-Qbit states $|\phi\rangle$ and $|\psi\rangle$, written in the form $\langle\phi|\psi\rangle$. In more conventional vector-space notation one would talk of an inner product between vectors ϕ and ψ, written in the form (ϕ, ψ).[21] The inner product is a number satisfying the usual rules for inner products in a vector space with complex scalars:

$$\langle\psi|\phi\rangle = \langle\phi|\psi\rangle^*,$$

$$\text{if } |\chi\rangle = \alpha|\psi\rangle + \beta|\lambda\rangle \text{ then } \langle\phi|\chi\rangle = \alpha\langle\phi|\psi\rangle + \beta\langle\phi|\lambda\rangle,$$

$$\langle\phi|\phi\rangle > 0, \quad |\phi\rangle \neq 0. \tag{1.75}$$

Since the two 1-Qbit states $|0\rangle$ and $|1\rangle$ are orthogonal unit vectors, we have

$$\langle 0|0\rangle = \langle 1|1\rangle = 1, \quad \langle 1|0\rangle = \langle 0|1\rangle = 0. \tag{1.76}$$

For the n-Qbit computational basis states $|x\rangle$ $(0 \leq x < 2^n)$ we have

$$\langle x|y\rangle = 0, \ x \neq y; \quad = 1, \ x = y, \tag{1.77}$$

[21] Some mathematicians find it ludicrous that quantum physicists insist on drawing extra lines "|" and "⟩" around all their vectors. It is, however, no sillier than the once widespread practice of drawing little arrows ($\rightarrow$) above all vectors. The application to Qbits illustrates the power of the notation, since what comes between the "|" and the "⟩" need not look like a vector at all, as in $|01101\rangle$.

since the inner product of two such n-fold tensor product is just the ordinary product of the n single-Qbit inner products.

It is often useful to think of the inner product $\langle\phi|\psi\rangle$ as a linear functional associated with the vector $|\phi\rangle$ that takes vectors $|\psi\rangle$ into complex numbers. Dirac gave the functional associated with $|\phi\rangle$ the name

$$\big(|\phi\rangle\big)^\dagger \text{ or } \langle\phi|, \tag{1.78}$$

so that the inner product $\langle\phi|\psi\rangle$ could actually be viewed as a compact expression of the functional $\langle\phi|$ acting on the vector $|\psi\rangle$:

$$\langle\phi|\psi\rangle = \langle\phi|\big(|\psi\rangle\big). \tag{1.79}$$

It is a standard result of linear algebra that the set of all such linear functionals on the original vector space is itself a vector space of the same dimension, known as the dual space. Dirac introduced the cloying but universal (among physicists) nomenclature of calling vectors like $|\psi\rangle$ in the original space *ket-vectors* or *kets*, and vectors like $\langle\phi|$ in the dual space *bra-vectors* or *bras*. This terminology was inspired (I'm not making this up) by the fact that the notation for the inner product (1.79) encloses the names of the two vectors between the bra[c]kets $\langle\ \ \rangle$.

The duals $\big(|x\rangle\big)^\dagger = \langle x|$ of the computational basis vectors are defined by linearity and their action (1.77) on the computational basis. To make the rules (1.75) for the inner product come out right we must have

$$\big(\alpha|\psi\rangle + \beta|\phi\rangle\big)^\dagger = \alpha^*\langle\psi| + \beta^*\langle\phi|, \tag{1.80}$$

which defines the dual for any superposition of computational basis vectors.

If you feel more comfortable with components, note that if we represent a general ket

$$|\psi\rangle = \sum_{x=0}^{2^{n-1}} \alpha_x |x\rangle_n \tag{1.81}$$

by the column vector

$$\begin{pmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_2 \\ \vdots \end{pmatrix} \tag{1.82}$$

then the associated bra $\langle\psi|$ is represented by the row vector

$$\big(\alpha_0^*\ \alpha_1^*\ \alpha_2^* \ldots\big), \tag{1.83}$$

so that the inner product of $|\psi\rangle$ with

$$|\phi\rangle = \sum_{x=0}^{2^{n-1}} \beta_x |x\rangle_n \tag{1.84}$$

is just the ordinary matrix product:

$$\langle\phi|\psi\rangle = \begin{pmatrix}\beta_0^* & \beta_1^* & \beta_2^* \ldots\end{pmatrix} \begin{pmatrix}\alpha_0 \\ \alpha_1 \\ \alpha_2 \\ \vdots\end{pmatrix} = \sum \beta_x^* \alpha_x. \tag{1.85}$$

If $\mathbf{A}$ is a general linear operator on kets, one also defines its action on bras by

$$\langle\psi|\mathbf{A} = \big(\mathbf{A}^\dagger|\psi\rangle\big)^\dagger. \tag{1.86}$$

It is a useful, if somewhat irritating, exercise in Dirac notation to prove that as so defined $\mathbf{A}$ is indeed linear on bras. Note that this definition extends the associative law to the three-fold product $\langle\psi|\mathbf{A}|\phi\rangle$ which can be interpreted, in more conventional notation, as being either $(\psi, \mathbf{A}\phi)$ or $(\mathbf{A}^\dagger\psi, \phi)$. As a famous teacher of mine once put it, "you can think of $\mathbf{A}$ as acting either to the right or to the left in $\langle\psi|\mathbf{A}|\phi\rangle$."

Relations like these permit us to extend to combinations of states and operators the general rule for operators that the adjoint of the adjoint is the original object, and the adjoint of a product is the product of the adjoints in the opposite order. Since the adjoint of a pure number (i.e. the operator which just multiplies any state by that number) is just the complex conjugate of the number, one has

$$\langle\psi|\mathbf{A}|\phi\rangle^* = \langle\psi|\mathbf{A}|\phi\rangle^\dagger = \langle\phi|\mathbf{A}^\dagger|\psi\rangle, \tag{1.87}$$

which gives us back the matrix definition of the adjoint as the complex conjugate of the transposed matrix.

One can also associate an outer product $|\psi\rangle\langle\phi|$ with a bra and a ket, defined as a linear operator on kets $|\chi\rangle$ satisfying

$$\big(|\psi\rangle\langle\phi|\big)|\chi\rangle = |\psi\rangle\big(\langle\phi|\chi\rangle\big). \tag{1.88}$$

If $|\psi\rangle$ and $|\phi\rangle$ have expansions (1.81) and (1.84) then the matrix expressing their outer product in the computational basis is given by the ordinary matrix product

$$\begin{pmatrix}\alpha_0 \\ \alpha_1 \\ \alpha_2 \\ \vdots\end{pmatrix} \begin{pmatrix}\beta_0^* & \beta_1^* & \beta_2^* \ldots\end{pmatrix}, \tag{1.89}$$

whose (x-y)th element is $M_{xy} = \alpha_x \beta_y^*$. Note that here (and almost everywhere else) Dirac notation eliminates the need for playing games with matrices, since the value of the matrix elements fall directly out of the notation. For example, $\langle x|\psi\rangle\langle\phi|y\rangle$ can be interpreted either as the matrix element

$$M_{xy} = \langle x|\big(|\psi\rangle\langle\phi|\big)|y\rangle \tag{1.90}$$

of the outer-product operator $M = |\psi\rangle\langle\phi|$ or, equally well, as the product of two complex numbers,

$$\big(\langle x|\psi\rangle\big)\big(\langle\phi|y\rangle\big). \tag{1.91}$$

The latter form (and the expansions (1.81) and (1.84) of $|\psi\rangle$ and $|\phi\rangle$ in the computational basis) gives us directly $\alpha_x\beta_y^*$. This is a good example of the primary point of Dirac notation: it has many built in ambiguities, but it is designed so that any way you chose to resolve those ambiguities is correct. In this way elementary little theorems become consequences of the notation. Mathematicians tend to loathe Dirac notation, because it prevents them from making distinctions they consider important. Physicists love Dirac notation, because they are always forgetting that such distinctions exist and the notation liberates them from having to remember.

An important special case of the outer product is the operator

$$\mathbf{P}_\psi = |\psi\rangle\langle\psi| \tag{1.92}$$

(not to be confused with the classical permutation operators $\mathbf{P}$). This is just the projection operator on the state $|\psi\rangle$. If a set of vectors $|\psi_i\rangle$, $i = 1 \ldots n$ constitute a complete orthonormal set, then the sum of the projections on each of them is just the unit operator:

$$\mathbf{1} = \sum_{i=1}^{n} |\psi_i\rangle\langle\psi_i|. \tag{1.93}$$

This trivial identity can be surprisingly useful. Letting both sides act on an arbitrary vector $|\phi\rangle$, for example, it tells us that the amplitudes we need to expand $|\phi\rangle$ in the basis given by the $|\psi\rangle_i$ are $\langle\psi_i|\phi\rangle$:

$$|\phi\rangle = \sum_{i=1}^{n} |\psi_i\rangle\langle\psi_i|\phi\rangle. \tag{1.94}$$

Sandwiching an arbitrary operator $\mathbf{A}$ between two such expansions of the identity, tells us how to expand $\mathbf{A}$ in terms of its matrix elements $A_{ij} = \langle\psi_i|A|\psi_j\rangle$ and the n^2 dimensional operator basis $|\psi_i\rangle\langle\psi_j|$ for the whole algebra of operators:

$$\mathbf{A} = \mathbf{1A1} = \sum_{i,j=1}^{n} |\psi_i\rangle\langle\psi_i|A|\psi_j\rangle\langle\psi_j| = \sum_{i,j=1}^{n} \big(\langle\psi_i|A|\psi_j\rangle\big)|\psi_i\rangle\langle\psi_j|\,. \tag{1.95}$$

The Born rule, relating the amplitudes in the expansion (1.58) of a general 1-Qbit state $|\psi\rangle$ to the probabilities of measuring 0 or 1 is often stated in terms of inner products: If a Qbit is in a state $|\psi\rangle$ then the probabilities of a measurement of the Qbit giving 0 or 1 are given by

$$p_\psi(0) = |\langle 0|\psi\rangle|^2, \quad p_\psi(1) = |\langle 1|\psi\rangle|^2. \tag{1.96}$$

This can also be written in terms of the projection operator $\mathbf{P}_\psi = |\psi\rangle\langle\psi|$ as

$$p_\psi(0) = \langle 0|\mathbf{P}_\psi|0\rangle, \quad p_\psi(1) = \langle 1|\mathbf{P}_\psi|1\rangle. \tag{1.97}$$

The rule can also be stated in terms of the projection operators $\mathbf{P}_0 = |0\rangle\langle 0|$ and $\mathbf{P}_1 = |1\rangle\langle 1|$ as

$$p_\psi(0) = \langle\psi|\mathbf{P}_0|\psi\rangle, \quad p_\psi(1) = \langle\psi|\mathbf{P}_1|\psi\rangle. \tag{1.98}$$

More generally, if $|\Psi\rangle$ is the state of n Qbits, then the probability of a measurement giving the result x ($0 \le x < 2^n$) is

$$p_\Psi(x) = |\langle x|\Psi\rangle|^2 = \langle x|\mathbf{P}_\Psi|x\rangle = \langle\Psi|\mathbf{P}_x|\Psi\rangle, \tag{1.99}$$

where $\mathbf{P}_\Psi = |\Psi\rangle\langle\Psi|$ and $\mathbf{P}_x = |x\rangle\langle x|$.

A2. Structure of the general 1-Qbit unitary transformation

I describe here some relations between Pauli matrices, unitary transformations, and real-space rotations. They are of fundamental importance in many applications of quantum mechanics, and are an essential part of the intellectual equipment of anybody wanting a deep understanding of 3-dimensional rotations. Occasionally they can be useful in applications to quantum computation.

The unit operator $\mathbf{1}$ and the three Pauli matrices (1.49) form a basis for the four dimensional algebra of two dimensional matrices: any two-dimensional matrix has a unique expansion of the form[22,23]

$$\mathbf{u} = u_0 + \mathbf{u}\cdot\sigma \tag{1.100}$$

[22] Please take care to distinguish between boldface $\mathbf{u}$, a 3-component vector, and boldface sans serif $\mathbf{u}$, an operator on two-dimensional vectors. You do not need a magnifying glass to distinguish them; it is always evident from the context.

[23] While in most of these notes we follow the computer science notation, calling σ_x, σ_y, and σ_z by their computer-science aliases, $\mathbf{X}$, $i\mathbf{Y}$, and $\mathbf{Z}$, the asymmetry introduced by the non-hermitian matrix $\mathbf{Y}$ makes the analysis that follows awkward and unattractive, so we stick here with the physicists' notation. Note also the physicists' practice — a generalization of their habit of not distinguishing notationally between the unit operator $\mathbf{1}$ and the scalar 1 — of omitting the explicit occurrence of $\mathbf{1}$ from $u_0\mathbf{1}$. In justification of what may strike you as an abominable practice, I point out that it is not so much a promotion of scalars to 2×2 matrices, as it is a demotion of the three Pauli matrices from 2×2 matrices to generalized (anti-commuting) scalars. It is quite analogous to regarding the ordinary complex numbers as an extension of the real numbers, rather than as real linear combinations of the two-dimensional matrices $\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$ and $\begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$.

for some complex number u_0 and 3-vector $\mathbf{u}$ with complex components u_x, u_y, and u_z. Here σ, unadorned by any subscript, represents the "3-vector" whose components are the Pauli matrices σ_x, σ_y, and σ_z given in (1.49), so in expanded form (1.100) reads

$$\mathbf{u} = u_0\mathbf{1} + u_x\sigma_x + u_y\sigma_y + u_z\sigma_z = \begin{pmatrix} u_0 + u_z & u_x - iu_y \\ u_x + iu_y & u_0 - u_z \end{pmatrix}. \tag{1.101}$$

As what follows demonstrates, however, it is invariably simpler to use the form (1.100) together with the multiplication rule (1.56), rather than dealing explicitly with 2-dimensional matrices.

We now impose on (1.100) the condition (1.65) that $\mathbf{u}$ be unitary. Since any unitary matrix remains unitary if it is multiplied by an overall multiplicative phase factor $e^{i\theta}$ with θ real, we can require u_0 to be real and arrive at a form which is general except for such an overall phase factor. Since the Pauli matrices are hermitian, we then have

$$\mathbf{u}^\dagger = u_0 + \mathbf{u}^* \cdot \sigma. \tag{1.102}$$

The rule (1.56) now tells us that for $\mathbf{u}$ to be unitary we must have

$$0 = 1 - \mathbf{u}^\dagger\mathbf{u} = 1 - u_0^2 - \mathbf{u}^* \cdot \mathbf{u} - \big(u_0(\mathbf{u} + \mathbf{u}^*) + i\mathbf{u}^* \times \mathbf{u}\big) \cdot \sigma. \tag{1.103}$$

Since $\mathbf{1}$, σ_x, σ_y, and σ_z, are linearly independent in the 4-dimensional algebra of 1-Qbit operators, the coefficients of all four in (1.103) must vanish and we have

$$1 = u_0^2 + \mathbf{u}^* \cdot \mathbf{u}, \quad 0 = u_0(\mathbf{u} + \mathbf{u}^*) + i\mathbf{u}^* \times \mathbf{u}. \tag{1.104}$$

The second of these requires the real and imaginary parts of the vector $\mathbf{u}$ to satisfy

$$u_0 \mathrm{Re}\,\mathbf{u} = \mathrm{Re}\,\mathbf{u} \times \mathrm{Im}\,\mathbf{u}. \tag{1.105}$$

If $u_0 \neq 0$, it follows from (1.105) that Re$\mathbf{u}\cdot$Re$\mathbf{u}$ =0, so Re$\mathbf{u}$ = 0, and the vector $\mathbf{u}$ must be i times a real vector $\mathbf{v}$. On the other hand if $u_0 = 0$ then (1.105) requires the real and imaginary parts of $\mathbf{u}$ to be parallel vectors, so that $\mathbf{u}$ itself is just a complex multiple of a real vector. But if $u_0 = 0$ we retain the freedom to pick the overall phase of the operator $\mathbf{u}$, which we can chose to make the vector $\mathbf{u}$ purely imaginary. So whether or not $u_0 = 0$, the general form for a two-dimensional unitary $\mathbf{u}$ is, to within an overall phase factor,

$$\mathbf{u} = u_0 + i\mathbf{v} \cdot \sigma, \tag{1.106}$$

where u_0 is a real number, $\mathbf{v}$ is a real vector, and, from the first of (1.104),

$$u_0^2 + \mathbf{v} \cdot \mathbf{v} = 1. \tag{1.107}$$

The identity (1.107) is ensured by parametrizing u_0 and $\mathbf{v}$ in terms of a real unit vector[24] $\mathbf{n}$ parallel to $\mathbf{v}$, and a real angle γ so that

$$\mathbf{u} = \cos\gamma + i(\mathbf{n}\cdot\sigma)\sin\gamma. \tag{1.108}$$

An alternative way of writing (1.108) is

$$\mathbf{u} = e^{i\gamma\mathbf{n}\cdot\sigma}. \tag{1.109}$$

(This follows from the forms of the power series expansions of the exponential, sine, and cosine, together with the fact that $(\mathbf{n}\cdot\sigma)^2 = 1$ for any unit vector $\mathbf{n}$ as a special case of (1.56). The argument is the same as the argument that $e^{i\varphi} = \cos\varphi + i\sin\varphi$ for any real number φ.)

A remarkable connection between these 2-dimensional unitary matrices and ordinary three dimensional rotations emerges from the fact that each of the three Pauli matrices (1.49) has zero trace,[25] and the fact[26] that the operator unitary transformation

$$\mathsf{A} \to \mathsf{u}\mathsf{A}\mathsf{u}^\dagger \tag{1.110}$$

preserves the trace of A.

Note first that if $\mathbf{a}$ is a real vector then $\mathbf{u}(\mathbf{a}\cdot\sigma)\mathbf{u}^\dagger$ is hermitian and can therefore be expressed as a linear combination of $\mathbf{1}$ and the three Pauli matrices with real coefficients. Since σ_x, σ_y, and σ_z all have zero trace, so does $\mathbf{a}\cdot\sigma$ and therefore so does $\mathbf{u}(\mathbf{a}\cdot\sigma)\mathbf{u}^\dagger$. Its expansion as a linear combination of $\mathbf{1}$ and the three Pauli matrices must therefore be of the form $\overline{\mathbf{a}}\cdot\sigma$ for some real vector $\overline{\mathbf{a}}$ (since $\mathbf{1}$ alone among the four has non-zero trace):

$$\mathbf{u}(\mathbf{a}\cdot\sigma)\mathbf{u}^\dagger = \overline{\mathbf{a}}\cdot\sigma. \tag{1.111}$$

It follows that

$$\mathbf{u}(\mathbf{a}\cdot\sigma)(\mathbf{b}\cdot\sigma)\mathbf{u}^\dagger = \big(\mathbf{u}(\mathbf{a}\cdot\sigma)\mathbf{u}^\dagger\big)\big(\mathbf{u}(\mathbf{b}\cdot\sigma)\mathbf{u}^\dagger\big) = \big(\overline{\mathbf{a}}\cdot\sigma\big)\big(\overline{\mathbf{b}}\cdot\sigma\big). \tag{1.112}$$

Since unitary transformations preserve the trace,

$$\mathrm{Tr}(\mathbf{a}\cdot\sigma)(\mathbf{b}\cdot\sigma) = \mathrm{Tr}(\overline{\mathbf{a}}\cdot\sigma)\big(\overline{\mathbf{b}}\cdot\sigma\big). \tag{1.113}$$

[24] Do not confuse boldface $\mathbf{n}$, the real unit 3-vector, with boldface sans serif $\mathbf{n}$, the 1-qubit number operator.

[25] The trace of a matrix is the sum of its diagonal elements.

[26] This is a consequence of the (easily verified) fact that the trace of a product of two operators is independent of the order in which the operators are multiplied, even though the operators may not commute.

Hence, from (1.56),

$$\overline{\mathbf{a}} \cdot \overline{\mathbf{b}} = \mathbf{a} \cdot \mathbf{b}. \tag{1.114}$$

But the most general real, linear,[27] inner-product-preserving transformation on real 3-vectors is a rotation. Consequently the transformation from real 3-vectors $\mathbf{a}$ to real 3-vectors $\overline{\mathbf{a}}$ induced by any 2-dimensional unitary $\mathbf{u}$ through (1.111) is a rotation:

$$\overline{\mathbf{a}} = \mathbf{R_u}\,\mathbf{a}. \tag{1.115}$$

Furthermore, by applying the (unitary) product $\mathbf{uv}$ of two unitary transformations in two steps,

$$(\mathbf{uv})(\mathbf{a}\cdot\sigma)(\mathbf{uv})^\dagger = \mathbf{u}\big(\mathbf{v}(\mathbf{a}\cdot\sigma)\mathbf{v}^\dagger\big)\mathbf{u}^\dagger = \mathbf{u}\big(R_\mathbf{v}\mathbf{a}\cdot\sigma\big)\mathbf{u}^\dagger = \mathbf{R_u}\mathbf{R_v}\mathbf{a}\cdot\sigma, \tag{1.116}$$

we deduce that

$$\mathbf{R_{uv}} = \mathbf{R_u}\mathbf{R_v}. \tag{1.117}$$

Thus the association of three-dimensional rotations with two-dimensional unitary matrices preserves the multiplicative structure of the rotation group: the rotation associated with the product of two unitary transformations is the product of the two associated rotations.

Which rotation is associated with which unitary transformation? To answer this note first that when the vector $\mathbf{a}$ in (1.111) is taken to be the vector $\mathbf{n}$ appearing in $\mathbf{u}$ (in (1.108) or (1.109)) then $\overline{\mathbf{n}} = \mathbf{n}$, since $\mathbf{u}$ then commutes with $\mathbf{n}\cdot\sigma$. Therefore $\mathbf{n}$ is along the axis of the rotation associated with $\mathbf{u} = e^{i\gamma\mathbf{n}\cdot\sigma}$. To determine the angle θ of that rotation let $\mathbf{m}$ be any unit vector perpendicular to the axis $\mathbf{n}$, so that

$$\cos\theta = \mathbf{m}\cdot\overline{\mathbf{m}}. \tag{1.118}$$

We then have

$$\begin{aligned}
\cos\theta &= \tfrac{1}{2}\mathrm{Tr}\big((\mathbf{m}\cdot\sigma)(\overline{\mathbf{m}}\cdot\sigma)\big)\\
&= \tfrac{1}{2}\mathrm{Tr}\big((\mathbf{m}\cdot\sigma)(\cos\gamma + i\sin\gamma\ \mathbf{n}\cdot\sigma)(\mathbf{m}\cdot\sigma)(\cos\gamma - i\sin\gamma\ \mathbf{n}\cdot\sigma)\big)\\
&= \tfrac{1}{2}\mathrm{Tr}\big((\cos\gamma\mathbf{m} - \sin\gamma\mathbf{m}\times\mathbf{n})\cdot\sigma)(\cos\gamma\mathbf{m} + \sin\gamma\mathbf{m}\times\mathbf{n})\cdot\sigma)\big)\\
&= \cos^2\gamma - \sin^2\gamma = \cos(2\gamma),
\end{aligned} \tag{1.119}$$

where we have made repeated use of (1.56) and the fact that $\mathbf{m}\cdot\mathbf{n} = 0$. So the unitary matrix (1.109) or (1.108) is associated with a rotation about the axis $\mathbf{n}$ through the angle 2γ. Since the identity rotation is associated both with $\mathbf{u} = \mathbf{1}$ and $\mathbf{u} = -\mathbf{1}$, the correspondence between these unitary matrices and three dimensional proper rotations is a 2-to-1 homomorphism. It is useful to introduce the notation $\mathbf{u}(\mathbf{n},\theta)$ for the 1-Qbit unitary transformation associated with the rotation $\mathbf{R}(\mathbf{n},\theta)$ about the axis $\mathbf{n}$ through the angle θ:

$$\mathbf{u}(\mathbf{n},\theta) = e^{i(\theta/2)(\mathbf{n}\cdot\sigma)} = \cos\tfrac{1}{2}\theta + i(\mathbf{n}\cdot\sigma)\sin\tfrac{1}{2}\theta. \tag{1.120}$$

[27] It follows directly from (1.111) that $\overline{\mathbf{a}+\mathbf{b}} = \overline{\mathbf{a}} + \overline{\mathbf{b}}$ and $\overline{\lambda\mathbf{a}} = \lambda\overline{\mathbf{a}}$.

The 3-dimensional rotations arrived at in this way are all *proper* (i.e. they preserve rather than invert handedness) because they can all be continuously connected to the identity. *Any* proper rotation can be associated with a $\mathbf{u}$, and in just two different ways ($\mathbf{u}$ and $-\mathbf{u}$ clearly being associated with the same rotation). The choice of phase leading to the general form (1.106) with real u_0 can be imposed by requiring that the determinant of $\mathbf{u}$ must be 1, so in mathematical language the 2-to-1 homomorphism is from the group SU(2) of unimodular unitary 2-dimensional matrices to the group SO(3) of proper 3-dimensional rotations.

Although all this may strike you as tediously abstract formalism, it is actually extremely useful at a very practical level, in reducing some highly nontrivial 3-dimensional geometry to elementary algebra, just as Euler's relation $e^{i\phi} = \cos\phi + i\sin\phi$ reduces some nontrivial 2-dimensional trigonometry to simple algebra. Suppose, for example, you combine a rotation through an angle α about an axis given by the unit vector $\mathbf{a}$ with a rotation through β about $\mathbf{b}$. The result, of course, is a single rotation. What is its angle γ and axis $\mathbf{c}$? Answering this question can be a nasty exercise in three-dimensional geometry. But to answer it using the Pauli matrices you only have to note that $\mathbf{u}(\mathbf{c},\gamma) = \mathbf{u}(\mathbf{a},\alpha)\mathbf{u}(\mathbf{b},\beta)$, i.e.

$$\cos\tfrac{1}{2}\gamma + i(\mathbf{c}\cdot\sigma)\sin\tfrac{1}{2}\gamma = \big(\cos\tfrac{1}{2}\alpha + i(\mathbf{a}\cdot\sigma)\sin\tfrac{1}{2}\alpha\big)\big(\cos\tfrac{1}{2}\beta + i(\mathbf{b}\cdot\sigma)\sin\tfrac{1}{2}\beta\big). \qquad (1.121)$$

Now multiply out the right side of (1.121), using (1.56). To get the angle γ take the trace of both sides (or identify the coefficients of unity) to find

$$\cos\tfrac{1}{2}\gamma = \cos\tfrac{1}{2}\alpha\cos\tfrac{1}{2}\beta - (\mathbf{a}\cdot\mathbf{b})\sin\tfrac{1}{2}\alpha\sin\tfrac{1}{2}\beta. \qquad (1.122)$$

To get the axis $\mathbf{c}$, identify the vectors of coefficients of the Pauli matrices:

$$\sin\tfrac{1}{2}\gamma\ \mathbf{c} = \sin\tfrac{1}{2}\beta\cos\tfrac{1}{2}\alpha\ \mathbf{b} + \sin\tfrac{1}{2}\alpha\cos\tfrac{1}{2}\beta\ \mathbf{a} - \sin\tfrac{1}{2}\alpha\sin\tfrac{1}{2}\beta\,(\mathbf{a}\times\mathbf{b}). \qquad (1.123)$$

Note that (1.122) and (1.123) are trivially correct when $\mathbf{a}$ and $\mathbf{b}$ are parallel. A little geometrical thought reveals that they are also correct when α and β are both 180°. To try to see geometrically why they are correct more generally is to acquire a deep appreciation for the remarkable power of the representation of 3-dimensional rotations in terms of 2-dimensional Pauli matrices.

A3. Structure of the general 1-Qbit state.

Let $|\phi\rangle$ be any 1-Qbit state, and let $|\psi\rangle$ be the orthogonal state (unique to within an overall phase), satisfying $\langle\psi|\phi\rangle = 0$. There is a unique unitary $\mathbf{u}$ taking the computational basis states $|0\rangle$ and $|1\rangle$ into $|\phi\rangle$ and $|\psi\rangle$.[28] Now the computational basis states are

[28] Since $|0\rangle$ and $|1\rangle$ are linearly independent there is a unique linear transformation taking them into $|\phi\rangle$ and $|\psi\rangle$. But since $|\phi\rangle$ and $|\psi\rangle$ are an orthonormal pair (as are $|0\rangle$ and $|1\rangle$) this linear transformation is easily verified to preserve the inner product of arbitrary pairs of states, so it is unitary.

eigenstates of the 1-Qbit number operator:

$$\mathbf{n}|x\rangle = x|x\rangle, \quad x = 0 \text{ or } 1, \tag{1.124}$$

where

$$\mathbf{n} = \tfrac{1}{2}(1 - \mathbf{Z}) = \tfrac{1}{2}(1 - \sigma_z) = \tfrac{1}{2}(1 - \mathbf{z} \cdot \sigma). \tag{1.125}$$

Since

$$|\phi\rangle = \mathbf{u}|0\rangle, \quad |\psi\rangle = \mathbf{u}|1\rangle, \tag{1.126}$$

the operator $\mathbf{n'} = \mathbf{u}\,\mathbf{n}\,\mathbf{u}^\dagger$ acts as a Qbit number operator on $|\phi\rangle$ and $|\psi\rangle$:

$$\mathbf{n'}|\phi\rangle = 0, \quad \mathbf{n'}|\psi\rangle = |\psi\rangle. \tag{1.127}$$

Since any 1-Qbit unitary transformation $\mathbf{u}$ is associated with a rotation $\mathbf{R}(\mathbf{m}, \theta)$, we have

$$\mathbf{n'} = \mathbf{u}\,\mathbf{n}\,\mathbf{u}^\dagger = \tfrac{1}{2}\big(1 - \mathbf{u}(\mathbf{z} \cdot \sigma)\mathbf{u}^\dagger\big) = \tfrac{1}{2}(1 - \mathbf{z'} \cdot \sigma), \tag{1.128}$$

where $\mathbf{z'} = R(\mathbf{m}, \theta)\mathbf{z}$.

Thus $\mathbf{n'}$, which functions as a number operator for the states $|\phi\rangle = \mathbf{u}(\mathbf{m}, \theta)|0\rangle = |0'\rangle$ and $|\psi\rangle = \mathbf{u}(\mathbf{m}, \theta)|1\rangle = |1'\rangle$ is constructed out of the component of the vector of operators σ along the direction $\mathbf{z'} = \mathbf{R}(\mathbf{m}, \theta)\mathbf{z}$ in exactly the same way that $\mathbf{n}$, the number operator for the computational basis states $|0\rangle$ and $|1\rangle$ is constructed out of the component along $\mathbf{z}$. This suggests that that there might be nothing special about the choice of $|0\rangle$ and $|1\rangle$ to form the computational basis states for each Qbit — that any pair of orthogonal states, $|0'\rangle = \mathbf{u}|0\rangle$ and $|1'\rangle = \mathbf{u}|1\rangle$ could serve as well. Furthermore it is at least a consistent possibility that to get the apparatus to measure the Qbits in this new basis we need do nothing more than than apply the rotation $\mathbf{R}$ associated with $\mathbf{u}$ to the apparatus that served to measure them in the original basis.

This physical possibility is realized by some, but by no means all, of the physical systems that have been proposed as possible embodiments of Qbits. It is realized for certain atomic magnets — also called *spins* — which have the property that when the magnetization of such a spin is measured along any given direction the magnet is found to be either maximally aligned along that direction, or maximally aligned opposite to that direction. These two possible outcomes for a particular direction — conventionally taken to be $\mathbf{z}$ — are associated with the values 0 and 1 for the Qbit. After such a measurement the spin is left in the state $|0\rangle$ or $|1\rangle$. Any other state $|\phi\rangle$ and its orthogonal partner $|\psi\rangle$ specify an alternative direction, along which the magnetization might have been measured, associated with an alternative scheme for reading out values for the Qbits. From this point of view the immensely greater set of possible states available to a Qbit than is available to a Cbit reflects the continuum of different ways one can read a Qbit (measuring its magnetization along any direction) as opposed to the single option available for reading a Cbit (finding out what value it actually has).

For Qbits that are not spins, the richness lies in the possibility of applying an *arbitrary* unitary transformation to each Qbit, before measuring it in the computational basis. What makes spins special is that applying the unitary transformation to the Qbits (which is not always that easy to arrange) can be replaced by straightforwardly applying the corresponding rotation to every one-Qbit measurement gate.

A4. An application of the formalism: "Spooky action at a distance"

Suppose Alice and Bob each has one member of a pair of Qbits, which have been prepared in the 2-Qbit state

$$|\Phi\rangle = \tfrac{1}{\sqrt{12}}\big(3|00\rangle + |01\rangle + |10\rangle - |11\rangle\big). \tag{1.129}$$

This state can also be written in terms of 1-Qbit Hadamard operators (1.42) as

$$|\Phi\rangle = \tfrac{1}{\sqrt{3}}\big(2|00\rangle - \mathbf{H}\otimes\mathbf{H}|11\rangle\big) = \tfrac{1}{\sqrt{3}}\big(2|00\rangle - \mathbf{H}_a\mathbf{H}_b|11\rangle\big), \tag{1.130}$$

where we take $\mathbf{H}_a$ to act on the left (Alice's) Qbit and $\mathbf{H}_b$ to act on the right (Bob's) Qbit.

It follows from (1.129) that If Alice and Bob each measures their Qbit, then there is a non-zero probability ($\frac{1}{12}$) that each will get the value 1. The following additional features of the state $|\Phi\rangle$ lead to a rather peculiar state of affairs:

(i) If Alice and Bob each applies a Hadamard to their Qbit then, since $\mathbf{H}^2 = 1$, the state (1.130) of the Qbits becomes

$$\mathbf{H}_a\mathbf{H}_b|\Phi\rangle = \tfrac{1}{\sqrt{3}}\big(2\mathbf{H}_a\mathbf{H}_b|00\rangle - |11\rangle\big) = \tfrac{1}{\sqrt{3}}\big(|00\rangle + |01\rangle + |10\rangle\big), \tag{1.131}$$

so if they measure their Qbits after each has applied a Hadamard, then there the probability that both will get the value 1 is zero.

(ii) If only Alice applies a Hadamard to her Qbit, then the state (1.130) of the Qbits becomes

$$\mathbf{H}_a|\Phi\rangle = \tfrac{1}{\sqrt{3}}\big(2\mathbf{H}_a|00\rangle - \mathbf{H}_b|11\rangle\big). \tag{1.132}$$

Since $\mathbf{H}_a|00\rangle = \big(\mathbf{H}|0\rangle\big)\otimes|0\rangle$, it is a linear combination of $|00\rangle$ and $|10\rangle$, and since $\mathbf{H}_b|11\rangle = |1\rangle\otimes\big(\mathbf{H}|1\rangle\big)$, it is a linear combination of $|10\rangle$ and $|11\rangle$. Thus the state $|01\rangle$ does not appear in the expansion of $\mathbf{H}_a|\Phi\rangle$ in computational basis states, so when the Qbits are subsequently measured, if Bob gets the value 1, there is zero probability that Alice will get the value 0.

(iii) If only Bob applies a Hadamard to his Qbit, then by exactly the same reasoning (except for the interchange of Alice and Bob), when the Qbits are subsequently measured if Alice gets the value 1, there is zero probability that Bob will get the value 0.

Taken all together, these probabilities for certain outcomes, depending on whether each does or does not apply a Hadamard gate before measuring, are quite strange. For

suppose that Alice and Bob each independently decide whether to apply a Hadamard to their qubit, before sending it through a measurement gate, by tossing a coin. There is then a small but non-zero probability ($\frac{1}{4} \times \frac{1}{12} = \frac{1}{48}$) that neither applies a Hadamard and both measurement gates show 1. In the one time in 48 that this happens, it is tempting to conclude that each Qbit was, even before the coins were tossed, *capable* of producing a 1 when directly subject to a measurement gate because, after all, each Qbit *did* produce a 1 when directly subject to a measurement gate.

But if Bob's Qbit had such a capability, then, in the absence of spooky interactions between Alice's Hadamard and Bob's Qbit, his Qbit surely would have retained that capability, even if Alice applied a pre-measurement Hadamard to her own Qbit. But if Bob's Qbit was indeed still capable of registering a 1 when measured directly, then Alice's Qbit must have been incapable of registering a 0 if measured after a Hadamard, since, as noted in (ii) above, when Alice applies a Hadamard before her measurement and Bob does not it is impossible for Alice's measurement to give 0 while Bob's gives 1.

By the same reasoning (interchanging Alice and Bob and referring to (iii) above) we conclude that Bob's Qbit must also have been incapable of registering a 0 when measured after a Hadamard.

So in each of the slightly more than 2% of the cases in which neither Bob nor Alice apply Hadamards and both their measurement gates register 1, we conclude that if the tosses of both coins had come out the other way and both had applied Hadamards before measuring, then neither Qbit could have registered 0 when measured: both would have had to register 1. But according to (i) above, the probability of both registering 1 after a Hadamard is zero.

Over the years[29] passions have run high on the significance of this. Some claim that it shows that the value Alice or Bob finds upon measuring *does* depend on whether or not the other, who could be far away, applies a Hadamard to his or her own Qbit before measuring. They call this "quantum nonlocality" or "spooky action at a distance"[30].

My own take on it is rather different. With any given pair of Qbits, each of Alice and Bob either does or does not apply a Hadamard prior to their measurement. Only one of the four possible cases is actually realized. The other three cases *do not happen.* In a deterministic world it can make sense to talk about what *would* have happened if things had been other than the way they actually were, since the hypothetical situation can entail unique subsequent behavior. But in the intrinsically nondeterminstic world that we actually inhabit, it makes no sense to infer, from what Bob's Qbit actually did, that it has a "capability" to do what it actually did, which it retains even in a hypothetical situation that did not, in fact, take place. To characterize the possible behavior of Bob's Qbit in the

[29] Although this particular argument was discovered by Lucien Hardy in the early 1990's, similar situations (where one has to work harder to unearth a paradox) have been known since a famous paper by John Bell appeared in 1964.

[30] This is a translation of Einstein's disparaging term *spukhafte Fernwirkungen.*

fictitious world requires more than just the irrelevance of Alice's decision whether or not to apply a Hadamard. It also requires that whatever it is that actually is relevant to Bob's outcome remains the same in both worlds and plays the same role in bringing about that outcome. But in the quantum world there is an irreducible randomness to such outcomes: nothing need play a role in bringing them about.[31]

The real lesson is that if one has a single pair of Qbits and various choices of gates to apply to them before sending them through a measurement gate, then it makes no sense to infer from the actual outcome of the measurement for the actual choice of gates, additional constraints (beyond those implied by the initial state of the Qbits) on the hypothetical outcomes of measurements in the fictional case in which one made a different choice of gates.

One can, however, let Alice and Bob repeatedly play this game with many different pairs of Qbits, always preparing the Qbits in the same initial 2-Qbit state (1.129). It is then entirely sensible to ask whether the *statistics* of the values Bob finds upon measuring his Qbit depend on whether Alice applied a Hadamard transform to her Qbit. For Alice and Bob can then accumulate a mass of data, and directly compare the statistics Bob got when Alice applied the Hadamard with those he got when she did not. If Bob got a different statistical distribution of readings depending on whether Alice did or did not apply a Hadamard to her faraway Qbit before she measured it, this would permit *non*spooky action at a distance which could actually be used to send messages. So it is important to note that Bob's *statistics* do not, in fact, depend on whether or not Alice applies a Hadamard.

We can show this under quite general conditions. Suppose n Qbits are divided into two subsets, each of which may be independently manipulated (i.e. subject to unitary transformations) prior to a measurement. Let the n_a Qbits on the left constitute one such group and the $n_b = n - n_a$ on the right, the other. Think of the first group as under the control of Alice and the second as belonging to Bob. If the n Qbits are always prepared in the state $|\Psi\rangle$, then if Alice and Bob separately measure their Qbits, the Born rule tells us that the joint probability $p(x, y)$ of Alice getting x and Bob, y, is

$$p_\Psi(x, y) = \langle\Psi|\mathbf{P}^a_x\mathbf{P}^b_y|\Psi\rangle, \tag{1.133}$$

where the projection operator $\mathbf{P}^a$ acts only on Alice's Qbits (i.e. it acts as the identity on Bob's) and $\mathbf{P}^b$, only on Bob's.

Suppose, now, that Alice acts on her Qbits with the unitary transformation $\mathbf{U}_a$ before making her measurement and Bob acts on his with $\mathbf{U}_b$. Then the state $|\Psi\rangle$ is changed into

$$|\Phi\rangle = \mathbf{U}_a\mathbf{U}_b|\Psi\rangle. \tag{1.134}$$

[31] Conscience requires me to report here the existence of a small but vocal deviant subculture of physicists, known as Bohmians, who maintain that there is a deterministic substructure, unfortunately inaccessible to us, that underlies quantum phenomena. Needless to say, all Bohmians are enthusiastic believers in real instantaneous action at a distance.

Now the probability of their measurements giving x and y, conditioned on their choices of unitary transformation, is

$$p_\Psi(x,y|\mathbf{U}_a,\mathbf{U}_b) = \langle\Phi|\mathbf{P}_x^a\mathbf{P}_y^b|\Phi\rangle = \langle\Psi|\mathbf{U}_b^\dagger\mathbf{U}_a^\dagger\big(\mathbf{P}_x^a\mathbf{P}_y^b\big)\mathbf{U}_a\mathbf{U}_b|\Psi\rangle$$

$$= \langle\Psi|(\mathbf{U}_a^\dagger\mathbf{P}_x^a\mathbf{U}_a)(\mathbf{U}_b^\dagger\mathbf{P}_y^b\mathbf{U}_b)|\Psi\rangle, \tag{1.135}$$

(where we have used the fact that all operators that act only on Alice's Qbits commute with all operators that act only on Bob's).

It follows from the fact that

$$\sum_x \mathbf{U}_a^\dagger\mathbf{P}_x^a\mathbf{U}_a = \mathbf{U}_a^\dagger\Big(\sum_x \mathbf{P}_x^a\Big)\mathbf{U}_a = \mathbf{U}_a^\dagger\mathbf{1}\mathbf{U}_a = 1, \tag{1.136}$$

that Bob's marginal statistics do not depend on what Alice chose to do to her own Qbits:

$$p_\Psi(y|\mathbf{U}_a\mathbf{U}_b) = \sum_x p_\Psi(x,y|\mathbf{U}_a,\mathbf{U}_b) = \langle\Psi|(\mathbf{U}_b^\dagger\mathbf{P}_y^b\mathbf{U}_b)|\Psi\rangle = p_\Psi(y|\mathbf{U}_b), \tag{1.137}$$

which does not depend on the particular unitary transformation $\mathbf{U}_a$ chosen by Alice. Therefore the statistics of the measurement outcomes for any group of Qbits, are not altered by anything done to other Qbits (provided, of course, the other Qbits do not subsequently interact with those in the original group, for example by the application of appropriate 2-Qbit gates).

A General Remark about the Figures

Most of the figures accompanying this and subsequent chapters are diagrams that enable one to see the action of a sequence of unitary gates on a collection of Qbits more transparently than through the corresponding formulae. In these circuit diagrams a Qbit is represented by a thin horizontal line (often called a wire) and a collection of Qbits by a thick horizontal line (sometimes called a bar). One-Qbit gates are represented by boxes or circles on a horizontal line; the multi-Qbit gates of interest are represented by such boxes, circles, or dots linked by vertical lines, or expanded so as to cover more than a single horizontal line..

The convention in the quantum computational literature is that the input Qbits appear as (ket) state vectors on the left side of such circuit diagrams, and the output Qbits, as state vectors on the right. If the state vectors are computational basis states like $|x\rangle$ one often omits the symbol $|\ \rangle$ and simply writes x. The advantage of putting the input on the left and the output on the right is that if one reads the diagram from left to right, as one does in European text, one encounters the gates in the order in which they act. A second advantage is that if one writes the diagram from left to right as one often does on blackboards, then one writes the gates in the order in which they act.

A significant disadvantage of the convention is that it contradicts the convention in physics notation that a symbol like $\mathbf{UVW}|\psi\rangle$ specifies the result of acting on the state $|\psi\rangle$ first with $\mathbf{W}$, then with $\mathbf{V}$, and finally with $\mathbf{U}$. So to represent

$$|\phi\rangle = \mathbf{UVW}|\psi\rangle \tag{1.138}$$

one draws a circuit diagram that looks like

$$|\psi\rangle \quad -\mathbf{W}-\mathbf{V}-\mathbf{U}- \quad |\phi\rangle, \tag{1.139}$$

with the operators in the diagram in the opposite sequence from those in the equation.

So beware of the possibility for confusion, arising from the fact that operators in circuit diagrams always appear in figures in the opposite sequence from the sequence they appear in on the page in the corresponding equations. Happily many of the diagrams we shall encounter are symmetric, and many of those that are not symmetric continue to be valid regardless of which direction you read them in.

Figures for Chapter 1

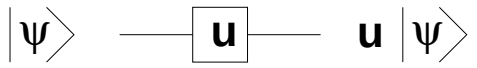

Figure 1.1. Circuit diagram representing the action on a single Qbit of the 1-Qbit gate **u**. Initially the Qbit is described by the input state $|\psi\rangle$ on the left. The thin line (wire) represents the subsequent history of the Qbit. After emerging from the box representing **u** the Qbit is described on the right by the final state $\mathbf{u}|\psi\rangle$.

Figure 1.2. Circuit diagram representing the action on n Qbits of the n-Qbit gate **U**. Initially the Qbits ares described by the input state $|\Psi\rangle$ on the left. The thick line (bar) represents the subsequent history of the Qbits. After emerging from the box representing **U** the Qbits are described on the right by the final state $\mathbf{U}|\Psi\rangle$.

Figure 1.3. Circuit diagram representing the action on n Qbits of two n-Qbit gates. Initially the Qbits ares described by the input state $|\Psi\rangle$ on the left. The thick line (bar) represents the subsequent history of the Qbits. First the are acted upon by the gate **V**, and then, by the gate **U**, emerging on the right in the final state $\mathbf{UV}|\Psi\rangle$. Note that the order in which the Qbits encounter unitary gates in the figure is opposite to the order in which the corresponding unitary symbols are written in the symbol for the final state on the right.

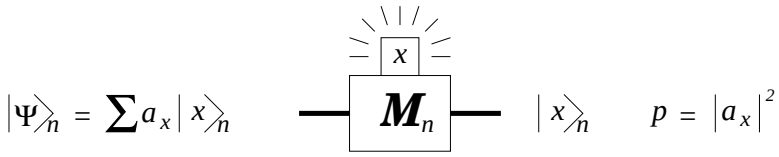

Figure 1.4. Circuit diagram representing an n-Qbit measurement gate. The Qbits are initially described by the n-Qbit state

$$|\Psi\rangle_n = \sum_{0 \le x < 2^n} a_x |x\rangle_n,$$

on the left. After the measurement gate $\mathbf{M}_n$ has acted, with probably $p = |a_x|^2$ it indicates an integer x, $0 \le x < 2^n$, and the Qbits are subsequently described by the state $|x\rangle_n$ on the right.

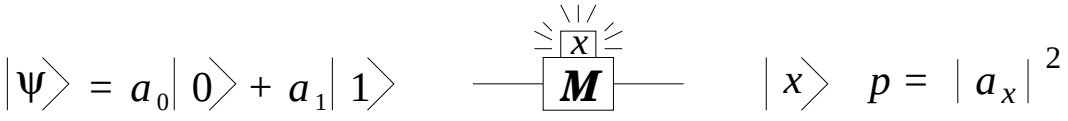

Figure 1.5. Special case of Figure 1.4: A 1-Qbit measurement gate. The reading x of the gate is either 0 or 1.

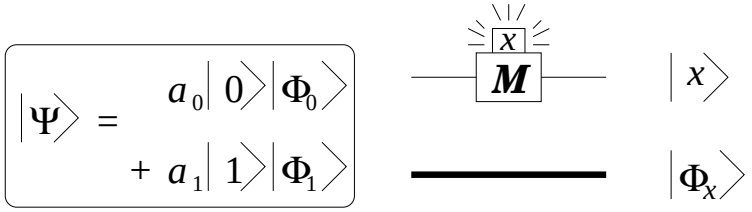

Figure 1.6. Action of a 1-Qbit measurement gate on a single one of n Qbits, according to the generalized Born rule. The initial state (on the left) is $|\Psi\rangle_n = a_0|0\rangle|\Phi_0\rangle_{n-1} + a_1|1\rangle|\Phi_1\rangle_{n-1}$. Only the single Qbit on the left is subject to a measurement gate.

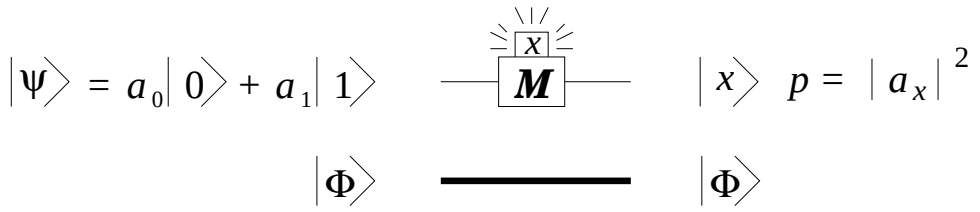

Figure 1.7. Simplification of Figure 1.6 when $|\Phi_0\rangle = |\Phi_1\rangle = |\Phi\rangle$. In this case the initial state on the right is just the product state $|\Psi\rangle_n = \big(a_o|0\rangle + a_1|1\rangle\big)|\Phi\rangle$, and the final state of the unmeasured Qbits continues to be $|\Phi\rangle$ regardless of the value of x indicated by the 1-Qbit measurement gate. Thus the unmeasured Qbits play no role at all: they are unentangled with the measured Qbit and described by the state Φ throughout the process. The 1-Qbit measurement gate acts on the measured Qbit exactly as it does in Figure 1.5 when no other Qbits are present, thereby demonstrating that the generalized Born rule of Figure 1.6 reduces to the ordinary Born rule, when the measured Qbit is unentangled with the others.

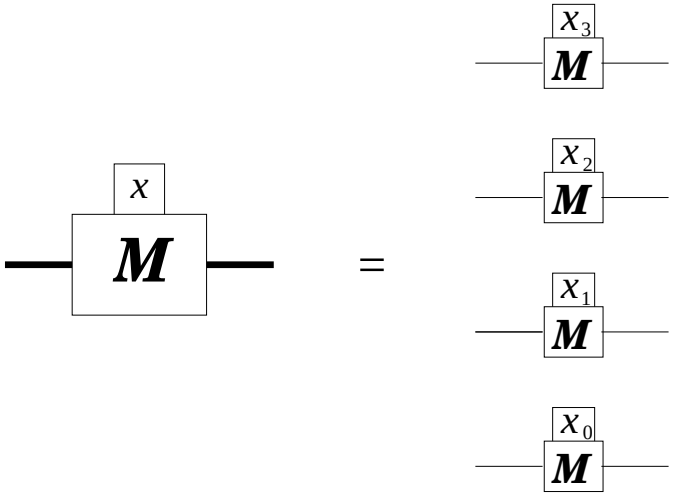

Figure 1.8. Constructing a 4-Qbit measurement gate out of four 1-Qbit measurement gates. The integer x has the binary expansion $x_3x_2x_1x_0$.

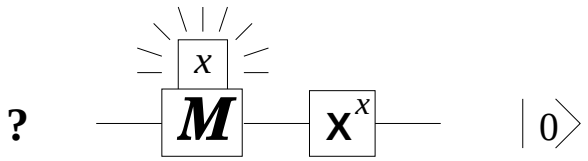

Figure 1.9. Using a 1-Qbit measurement gate to prepare an off-the-shelf Qbit so its associated state is $|0\rangle$. The input on the left is a Qbit in an unknown condition — i.e. nothing is known of its past history. After the measurement gate is applied, **X** is or is not applied, depending on whether the measurement gate indicates 1 or 0. The Qbit emerges (on the right) in the state $|0\rangle$.