Last revised 1/28/03

LECTURE NOTES ON QUANTUM COMPUTATION

Cornell University, Physics 481-681, CS 483; Spring, 2003

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 takes advantage of certain kinds of transformations in its internal state that the laws of quantum mechanics allow under very special circumstances.

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 a classical computer, introduce potentially catastrophic disruptions into the operation of a quantum computer. Such disastrous interactions include not only interactions with the external environment — air molecules bouncing off the physical systems that represent bits, or those systems absorbing a minute amount (a single quantum) of ambient radiant thermal energy — but also interactions between the computationally relevant degrees of freedom of such systems with irrelevant thermally excited degrees of freedom associated with their internal structure. Such interactions are said to result in "decoherence", which is death to a quantum computation.

What this means is that individual bits cannot be encoded in physical systems of macroscopic size (because they cannot be isolated from their own irrelevant internal degrees of freedom) but must be encoded in a very small number of quantum states of a system of atomic size (so that extra internal degrees of freedom do not come into play because they do not exist) which is decoupled from all of its surroundings except for the completely controlled couplings to the physical systems that encode other bits.

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 external interactions can actually be corrected, provided they occur at a sufficently low rate. While error correction is routine for classical bits, 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.

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 at a very deep level the theory of computation cannot be divorced from the physics of the devices that embody the computation, and (c) as a source of new examples to illustrate and illuminate the surprising 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. About a year ago I mentioned to a distinguished theoretical physicist (an authority on string theory and director of a great theoretical physics institute) 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 an actual physical system — 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 (and lose much wisdom, none of it — at least at this stage of the art — 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 glimpse 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, for the special task for which it is intended.

I should mention 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 behavior sometimes show up at the level of quantum computation. One of the major appeals of quantum computation for me is that it affords a new conceptual area 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 (as I easily could) letting it pass by unremarked upon and unnoticed. (One such example is described in Section A3 of the Appendix to this Chapter.)

We begin our survey of quantum computation with a minimalist introduction to the quantum theory (also called "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 this rather specialized topic — the physical manipulation of digital information — it is possible to give a characterization of how the quantum theory works which is quite concise

but nevertheless complete for this (limited) area of application.

While I assume here no prior familiarity with the quantum theory, 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. I hope that introducing and applying the unfamiliar nomenclature and notation of quantum mechanics in a familiar classical context will 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 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 also to use the term "bit" to describe such a two-state classical system, but this use of a single term to characterize both the abstract bit (0 or 1) and the physical system whose two states represent its two values is a potential source of confusion. To avoid such confusion in this Chapter I shall use the term *Cbit* ("C" for "classical") to describe the two-state physical system and *Qbit* for its quantum generalization. This terminology is inspired by Paul Dirac's early use of *c-number* and *q-number* to describe classical quantities and their quantum-mechanical generalizations. ("Cbit" and "Qbit" are preferable to "c-bit" and "q-bit" because the terms themselves often appear in hyphenated constructions.) Unfortunately the orthographically preposterous term *qubit* currently holds sway for the quantum system.[2] Although it honors

[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 the subject can be found in John Preskill's Caltech lecture notes, available at `http://www.theory.caltech.edu/people/preskill/ph229/`.

[2] *Qubit* seems first to have been used in print by Benjamin Schumacher, "Quantum

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 it has gained acceptance because it visually resembles an ancient English unit of distance, the homonymic *cubit*. To see its ungainliness with fresh eyes, 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 introductory exposition that follows, I use this unfashionable terminology in Chapter 1. In subsequent Chapters Cbits will play a very minor role, and I shall set aside my aesthetic principles, reverting to the accepted term *qubit* except when it is important to make distinctions between Qbits and Cbits.

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. So the two possible 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 $\mathbf{x}$ that represents the physical position of an object. I call it to your attention only because in the quantum case "state" refers *only* to the symbol — there is no internal property of the Qbit that it represents.)

Along the same lines, we shall characterize the states of the 5 Cbits representing 11001, for example, by the symbol

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

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,

$$|0\rangle|0\rangle, |0\rangle|1\rangle,\ |1\rangle|0\rangle,\ \text{or}|1\rangle|1\rangle, \tag{2}$$

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

$$|0\rangle|0\rangle|0\rangle,\ |0\rangle|0\rangle|1\rangle,\ |0\rangle|1\rangle|0\rangle,\ |0\rangle|1\rangle|1\rangle,\ |1\rangle|0\rangle|0\rangle,\ |1\rangle|0\rangle|1\rangle,\ |1\rangle|1\rangle|0\rangle,\ \text{or}\ |1\rangle|1\rangle|1\rangle, \tag{3}$$

and so on.

As (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:

$$|000\rangle,\ |001\rangle,\ |010\rangle,\ |011\rangle,\ |100\rangle,\ |101\rangle,\ |110\rangle,\ \text{or}\ |111\rangle. \tag{4}$$

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.

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 (3) or (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 (4) by the even shorter forms:

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

Note, though, that unlike the forms (3) and (4), the form (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:

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

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

$$|5 \text{ horizontal centimeters northeast}\rangle. \tag{7}$$

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.

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 affectation for Cbits, it is absolutely fundamental to the description of Qbits. Playing silly games with Cbits will, I hope, provide a painless (though possibly boring) way to become introduced to some of the quantum mechanical formalism in a familiar setting.

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

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

$$|0\rangle \longleftrightarrow \begin{pmatrix} 1 \\ 0 \end{pmatrix}. \;\; |1\rangle \longleftrightarrow \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \tag{8}$$

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

$$|00\rangle,\; |01\rangle,\; |10\rangle,\; |11\rangle. \tag{9}$$

The alternative notation for this basis,

$$|0\rangle|0\rangle,\; |0\rangle|1\rangle,\; |1\rangle|0\rangle,\; |1\rangle|1\rangle, \tag{10}$$

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

$$|0\rangle \otimes |0\rangle, \quad |0\rangle \otimes |1\rangle, \quad |1\rangle \otimes |0\rangle, \quad |1\rangle \otimes |1\rangle. \tag{11}$$

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 becomes the natural way to represent multi-Cbit states, since it leads to the obvious multi-Cbit generalization of the representation (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 for a three-fold tensor product:

$$\begin{pmatrix} x_0 \\ x_1 \end{pmatrix} \otimes \begin{pmatrix} y_0 \\ y_1 \end{pmatrix} \otimes \begin{pmatrix} z_0 \\ z_1 \end{pmatrix} \longleftrightarrow \begin{pmatrix} x_0y_0z_0 \\ x_0y_0z_1 \\ x_0y_1z_0 \\ x_0y_1z_1 \\ x_1y_0z_0 \\ x_1y_0z_1 \\ x_1y_1z_0 \\ x_1y_1z_1 \end{pmatrix}. \tag{12}$$

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

$$|5\rangle_3 = |101\rangle = |1\rangle|0\rangle|1\rangle \;\longleftrightarrow\; \begin{pmatrix} 0 \\ 1 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}. \tag{13}$$

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 occurs 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 (13). This is indeed the obvious multi-Cbit generalization of the column-vector form (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 given by the binary number that those n bits represent.

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 to its final form using only processes whose action can be inverted. There is an equally important *irreversible* part of the operation of a quantum computer called *measurement*, which extracts useful information from the Qbits after their state has acquired its final form. In a classical computer the extraction of information from the final state of the Cbits is so straightforward a procedure that it is rarely even described as part of the computational process (though it is, of course, a major concern for those who design digital displays or printers). All conceptually nontrivial operations in a classical computer occur in the process of transforming the initial state of the Cbits to its final form. Since quantum computers must do this part of their operation through reversible operations, only reversible operations on Cbits will be of interest to us here. (An example of an irreversible operation would be ERASE which forces the Cbit into the state $|0\rangle$ regardless of whether its initial state was $|0\rangle$ or $|1\rangle$. ERASE is irreversible because given only the final state and the fact that it was the output of 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*) $|0\rangle$ and $|1\rangle$:

$$\mathbf{X} : |x\rangle \to |\overline{x}\rangle, \quad \overline{1} = 0, \quad \overline{0} = 1. \tag{14}$$

It is reversible because it has an inverse: applying NOT a second time brings the Cbit back to its initial state. If we represent the states of the Cbit by the two orthogonal 2-vectors $|0\rangle$ and $|1\rangle$ then we can express NOT by a linear operator $\mathbf{X}$ on the 2-dimensional vector space, whose action on the column vectors (8) is given by the matrix

$$\mathbf{X} \longleftrightarrow \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}. \tag{15}$$

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 $\mathbf{1}$,

$$\mathbf{1} \longleftrightarrow \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \tag{16}$$

on its two-dimensional vector space.[4]

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

$$\mathbf{n}|x\rangle = x|x\rangle, \quad x = 0 \text{ or } 1; \tag{17}$$

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,

$$\overline{\mathbf{n}} = \mathbf{1} - \mathbf{n}, \tag{18}$$

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

$$\mathbf{n} \longleftrightarrow \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}, \quad \overline{\mathbf{n}} \longleftrightarrow \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}. \tag{19}$$

It follows directly from their definitions that

$$\mathbf{n}^2 = \mathbf{n}, \quad \overline{\mathbf{n}}^2 = \overline{\mathbf{n}}, \quad \mathbf{n}\overline{\mathbf{n}} = \overline{\mathbf{n}}\mathbf{n} = \mathbf{0}, \quad \mathbf{n} + \overline{\mathbf{n}} = \mathbf{1}. \tag{20}$$

We also have

$$\mathbf{n}\mathbf{X} = \mathbf{X}\overline{\mathbf{n}}, \quad \overline{\mathbf{n}}\mathbf{X} = \mathbf{X}\mathbf{n}, \tag{21}$$

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

$$\mathbf{X}^2 = \mathbf{1}: \tag{22}$$

the NOT operator is its own inverse. All the simple relations (20)-(22) also follow, as they must, from the matrix representations (15) and (19) for $\mathbf{X}$, $\mathbf{n}$, and $\overline{\mathbf{n}}$.

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

[4] 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 — particularly when writing on the blackboard. The same applies to the scalar 0, the zero vector $\mathbf{0}$, and the zero operator $\mathbf{0}$.

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[5]

$$\mathbf{S} = \mathbf{n} \otimes \mathbf{n} + \overline{\mathbf{n}} \otimes \overline{\mathbf{n}} + (\mathbf{X} \otimes \mathbf{X})(\mathbf{n} \otimes \overline{\mathbf{n}}) + (\mathbf{X} \otimes \mathbf{X})(\overline{\mathbf{n}} \otimes \mathbf{n}). \tag{23}$$

Here the tensor product $\otimes$ of two 1-bit operators[6] 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.

$$\mathbf{a} \otimes \mathbf{b}\ |x\rangle \otimes |y\rangle = \mathbf{a}|x\rangle \otimes \mathbf{b}|y\rangle, \tag{24}$$

from which it follows that[7]

$$(\mathbf{a} \otimes \mathbf{b})(\mathbf{c} \otimes \mathbf{d}) = (\mathbf{ac}) \otimes (\mathbf{bd}). \tag{25}$$

At the risk of belaboring the obvious, I note that (23) 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[8] (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 (24) acts only on the 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

$$\mathbf{1} \otimes \mathbf{1} \otimes \mathbf{a} \otimes \mathbf{1} \otimes \mathbf{b} \otimes \mathbf{1}. \tag{26}$$

[5] 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.

[6] 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.

[7] 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.

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

To avoid such typographical monstrosities we simplify (26) to

$$\mathbf{1} \otimes \mathbf{1} \otimes \mathbf{a} \otimes \mathbf{1} \otimes \mathbf{b} \otimes \mathbf{1} = \mathbf{a}_3\mathbf{b}_1 = \mathbf{b}_1\mathbf{a}_3, \tag{27}$$

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 (27) doesn't matter: 1-bit operators that act on different Cbits commute.

So with this convention[9], if the swap operator $\mathbf{S}$ acts on Cbits i and j, we can rewrite (23) as

$$\mathbf{S}_{ij} = \mathbf{n}_i\mathbf{n}_j + \overline{\mathbf{n}}_i\overline{\mathbf{n}}_j + (\mathbf{X}_i\mathbf{X}_j)(\mathbf{n}_i\overline{\mathbf{n}}_j + \overline{\mathbf{n}}_i\mathbf{n}_j). \tag{28}$$

To help you become more at home with this notation, you are invited to prove as an exercise in algebra, that using only the relations (20)-(22) and the fact that one-bit operators acting on different Cbits commute, it follows from (28) 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. Formally we can write

$$\mathbf{C}_{10}|x\rangle|y\rangle = |x\rangle|y \oplus x\rangle, \tag{29}$$

where $\oplus$ denotes addition modulo 2:

$$y \oplus 0 = y, \quad y \oplus 1 = \overline{y} = 1 - y. \tag{30}$$

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

$$\mathbf{S}_{ij} = \mathbf{C}_{ij}\mathbf{C}_{ji}\mathbf{C}_{ij}. \tag{31}$$

[9] 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 in this case) without reference to which Cbit it acts on.

This can be verified by repeated applications of (29). It can also be demonstrated using the algebraic approach. Note first that $\mathbf{C}$ can be expressed in terms of $\mathbf{n}$'s and $\mathbf{X}$'s by

$$\mathbf{C}_{ij} = \overline{\mathbf{n}}_i + \mathbf{X}_j \mathbf{n}_i, \tag{32}$$

since if the state of Cbit i is $|0\rangle$ only the first term acts which leaves Cbit j unchanged, but if the state of Cbit i is $|1\rangle$ only the second term acts and $\mathbf{X}_j$ flips Cbit j. If you substitute expressions of the form (32) for each of the three terms in (31), then with the help of (20)-(22) (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 (28).

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

$$\mathbf{Z} = \overline{\mathbf{n}} - \mathbf{n} \;\longleftrightarrow\; \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}. \tag{33}$$

This is a rather peculiar operator, since applied to $|1\rangle$ it produces $-|1\rangle$ and it is not at all clear what it means to multiply the vector representing the state of a Cbit by -1 (or any other number). Never mind. For now we are only going to use it combined with other operators to produce perfectly sensible transformations on Cbits. It follows from (21) (or from the matrix representations (15) and (33)) that the NOT operator $\mathbf{X}$ *anticommutes* with $\mathbf{Z}$:

$$\mathbf{ZX} = -\mathbf{XZ}. \tag{34}$$

Since $\overline{\mathbf{n}} + \mathbf{n} = \mathbf{1}$ we can use (33) to express $\overline{\mathbf{n}}$ and $\mathbf{n}$ in terms of $\mathbf{1}$ and $\mathbf{Z}$:

$$\mathbf{n} = \tfrac{1}{2}(\mathbf{1} - \mathbf{Z}), \quad \overline{\mathbf{n}} = \tfrac{1}{2}(\mathbf{1} + \mathbf{Z}). \tag{35}$$

Using this we can then rewrite the cNOT operator (32) in terms of $\mathbf{X}$ and $\mathbf{Z}$ operators:

$$\begin{aligned} \mathbf{C}_{ij} &= \tfrac{1}{2}\big(\mathbf{1} + \mathbf{Z}_i\big) + \tfrac{1}{2}\mathbf{X}_j\big(\mathbf{1} - \mathbf{Z}_i\big) \\ &= \tfrac{1}{2}\big(\mathbf{1} + \mathbf{X}_j\big) + \tfrac{1}{2}\mathbf{Z}_i\big(\mathbf{1} - \mathbf{X}_j\big). \end{aligned} \tag{36}$$

(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 (36) 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.*),

$$\mathbf{H} = \tfrac{1}{\sqrt{2}}(\mathbf{X} + \mathbf{Z}) \;\longleftrightarrow\; \tfrac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}. \tag{37}$$

This is another great work-horse of quantum computation. (Physicists should note here an unfortunate clash between computer-science and physics notations. Quantum physicists invariably use H to denote the Hamiltonian function (in classical mechanics) or operator (in quantum mechanics). Fortunately Hamiltonian operators, although of crucial importance in the design of quantum computers, play a fairly 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[10] that

$$\mathbf{H}^2 = \mathbf{1} \tag{38}$$

and that

$$\mathbf{HXH} = \mathbf{Z}, \quad \mathbf{HZH} = \mathbf{X}. \tag{39}$$

So $\mathbf{H}$ can be used in this way to interchange $\mathbf{X}$ and $\mathbf{Z}$ and we conclude from (36), (38), and (39), that

$$\mathbf{C}_{ji} = \big(\mathbf{H}_i\mathbf{H}_j\big)\mathbf{C}_{ij}\big(\mathbf{H}_i\mathbf{H}_j\big). \tag{40}$$

Of course the action of $\mathbf{H}$ on the state of a Cbit,

$$\mathbf{H}|0\rangle = \tfrac{1}{\sqrt{2}}(|0\rangle + |1\rangle), \quad \mathbf{H}|1\rangle = \tfrac{1}{\sqrt{2}}(|0\rangle - |1\rangle), \tag{41}$$

makes no sense at all. Nevertheless, when combined with other operations, as on the right side of (40), the Hadamard operations result in the perfectly sensible operation given on the left side.

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.

If you would like a matrix for the cNOT in the 4-dimensional 2-Qbit subspace note that if the control bit is on the 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

$$\begin{pmatrix} 1&0&0&0\\ 0&1&0&0\\ 0&0&0&1\\ 0&0&1&0 \end{pmatrix}. \tag{42}$$

If the control bit is on the right then it is $|01\rangle$ and $|11\rangle$ that are interchanged and the matrix is

$$\begin{pmatrix} 1&0&0&0\\ 0&0&0&1\\ 0&0&1&0\\ 0&1&0&0 \end{pmatrix}. \tag{43}$$

[10] This also follows, of course, from the matrix representation (37) of $\mathbf{H}$.

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

$$\begin{pmatrix} 1&0&0&0\\ 0&0&1&0\\ 0&1&0&0\\ 0&0&0&1 \end{pmatrix}. \tag{44}$$

It is, however, usually much more efficient in establishing various operator identities to learn to deal with them directly as operators, avoiding such computational-basis matrix representations.

Before turning to the quantum generalization of Cbits, let me show you, as a further exercise in treating operations on classical bits as linear operations on vectors, a curious way of rewriting the swap operator (28). The form has a rather more elegant structure than (28), though you may well regard it as an utterly perverse way to treat Cbits.

If we use (35) to reexpress each $\mathbf{n}$ and $\overline{\mathbf{n}}$ appearing in the swap operator (28) in terms of $\mathbf{Z}$, we find that

$$\mathbf{S}_{ij} = \tfrac{1}{2}(\mathbf{1} + \mathbf{Z}_i\mathbf{Z}_j) + \tfrac{1}{2}(\mathbf{X}_i\mathbf{X}_j)(\mathbf{1} - \mathbf{Z}_i\mathbf{Z}_j). \tag{45}$$

If we define

$$\mathbf{Y} = \mathbf{Z}\mathbf{X} = -\mathbf{X}\mathbf{Z} \;\longleftrightarrow\; \begin{pmatrix} 0&1\\ -1&0 \end{pmatrix}, \tag{46}$$

we get the more compact form

$$\mathbf{S}_{ij} = \tfrac{1}{2}(\mathbf{1} + \mathbf{X}_i\mathbf{X}_j - \mathbf{Y}_i\mathbf{Y}_j + \mathbf{Z}_i\mathbf{Z}_j). \tag{47}$$

We can get rid of the irritating minus sign that mars the elegance of (47) 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 notation much beloved by physicists:

$$\sigma_x = \mathbf{X} \;\longleftrightarrow\; \begin{pmatrix} 0&1\\ 1&0 \end{pmatrix},\ \sigma_y = -i\mathbf{Y} \;\longleftrightarrow\; \begin{pmatrix} 0&-i\\ i&0 \end{pmatrix},\ \sigma_z = \mathbf{Z} \;\longleftrightarrow\; \begin{pmatrix} 1&0\\ 0&-1 \end{pmatrix}, \tag{48}$$

in terms of which the swap operator becomes

$$\mathbf{S}_{ij} = \tfrac{1}{2}(1 + \sigma_x^{(i)}\sigma_x^{(j)} + \sigma_y^{(i)}\sigma_y^{(j)} + \sigma_z^{(i)}\sigma_z^{(j)}). \tag{49}$$

Physicists might note 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 (49). In addition the three σ matrices all square to unity,

$$\sigma_x^2 = \sigma_y^2 = \sigma_z^2 = 1, \tag{50}$$

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

$$\begin{aligned} \sigma_x\sigma_y &= -\sigma_y\sigma_x = i\sigma_z, \\ \sigma_y\sigma_z &= -\sigma_z\sigma_y = i\sigma_x, \\ \sigma_z\sigma_x &= -\sigma_x\sigma_z = i\sigma_y. \end{aligned} \tag{51}$$

Note that the three relations (51) differ only by cyclic permutations of x, y, and z.

All the relations (50) and (51) can be summarized in a single compact identity. Let $\mathbf{a}$ and $\mathbf{b}$ be two three-dimensional vectors,

$$\mathbf{a} = (a_x, a_y, a_z), \quad \mathbf{b} = (b_x, b_y, b_z), \tag{52}$$

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

$$\sigma = (\sigma_x, \sigma_y, \sigma_z), \tag{53}$$

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:

$$\mathbf{a} \cdot \sigma = a_x\sigma_x + a_y\sigma_y + a_z\sigma_z. \tag{54}$$

Then the relations (50) and (51) imply and are implied by the single identity

$$(\mathbf{a} \cdot \sigma)(\mathbf{b} \cdot \sigma) = \mathbf{a} \cdot \mathbf{b} + i(\mathbf{a} \times \mathbf{b}) \cdot \sigma, \tag{55}$$

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

Note that by building the i into the definition of σ_y we have also arranged so that all three of the σ matrices are hermitian.[11] 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[12]

$$A = a_0 + \mathbf{a} \cdot \sigma, \tag{56}$$

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

[12] 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.

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.[13] 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 be making extensive use of the Pauli matrices, disguised as $\mathbf{X}, \mathbf{Y}$, and $\mathbf{Z}$, when we turn to Qbits, as we now do.

C. Qbits and their states.

A Cbit is a pretty miserable specimen of a two-dimensional vector, since the only vectors with any classical meaning in the 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 $|1\rangle$. Qbits do not suffer from this limitation. The state $|\psi\rangle$ of 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 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

$$|\psi\rangle = \alpha_0|0\rangle + \alpha_1|1\rangle \ \longleftrightarrow \ \begin{pmatrix} \alpha_0 \\ \alpha_1 \end{pmatrix}, \tag{57}$$

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

$$|\alpha_0|^2 + |\alpha_1|^2 = 1. \tag{58}$$

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 convenience.

Just as the general state of a single Qbit is an arbitrary normalized superposition (57) of the two possible classical states, the general state $|\psi\rangle$ of two Qbits is an arbitrary

[13] Hamilton's quaternions i, j, k are represented by $i\sigma_x, i\sigma_y, i\sigma_z$. The connection between the Pauli matrices and three-dimensional rotations is developed in Section A1 of the Appendix to this Chapter.

normalized superposition of the four orthogonal classical states,

$$|\psi\rangle = \alpha_{00}|00\rangle + \alpha_{01}|01\rangle + \alpha_{10}|10\rangle + \alpha_{11}|11\rangle \;\longleftrightarrow\; \begin{pmatrix} \alpha_{00} \\ \alpha_{01} \\ \alpha_{10} \\ \alpha_{11} \end{pmatrix}, \tag{59}$$

with the complex amplitudes being constrained only by the normalization condition

$$|\alpha_{00}|^2 + |\alpha_{01}|^2 + |\alpha_{10}|^2 + |\alpha_{11}|^2 = 1. \tag{60}$$

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

$$|\psi\rangle = \sum_{0 \le x < 2^n} \alpha_x |x\rangle_n, \tag{61}$$

$$\sum_{0 \le x < 2^n} |\alpha_x|^2 = 1. \tag{62}$$

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,* though for many purposes *classical basis* would be a more appropriate term. 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,

$$\begin{aligned} |\psi\rangle \otimes |\phi\rangle =& \big(\alpha_0|0\rangle + \alpha_1|1\rangle\big) \otimes \big(\beta_0|0\rangle + \beta_1|1\rangle\big) \\ &= \alpha_0\beta_0|00\rangle + \alpha_0\beta_1|01\rangle + \alpha_1\beta_0|10\rangle + \alpha_1\beta_1|11\rangle \;\longleftrightarrow\; \begin{pmatrix} \alpha_0\beta_0 \\ \alpha_0\beta_1 \\ \alpha_1\beta_0 \\ \alpha_1\beta_1 \end{pmatrix}. \end{aligned} \tag{63}$$

Note that a *general* 2-Qbit state (59) is of the special form (63) if and only if $\alpha_{00}\alpha_{11} = \alpha_{01}\alpha_{10}$. Since the four amplitudes in (59) are constrained only by the normalization condition (60), this relation need not hold, and the general 2-Qbit state, unlike the general state of two Cbits, is *not* a product (63) 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, in contrast

to individual Cbits, cannot always be characterized as having individual states of their own.[14]

Such nonproduct states of two or more Qbits are called *entangled* states.[15] Entanglement is one of the most peculiar properties quantum bits can have. It can lead to some very strange behavior, an example of which 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 $\mathbf{X}$. A Qbit is considerably more versatile. The reversible operations that a quantum computer can perform upon single Qbit are represented by the action on the state of the Qbit by 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[16]

$$\mathbf{u}\mathbf{u}^\dagger = \mathbf{u}^\dagger\mathbf{u} = 1. \tag{64}$$

Since any unitary transformation has a unitary inverse, such actions of a quantum computer on a Qbit are reversible. The reason why these operations on Qbits *must* be reversible for a quantum computer to function effectively will emerge in Chapter II.

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 on n-Qbits in a given state is any linear transformation that takes unit vectors into unit vectors — i.e. any 2^n-dimensional unitary transformation $\mathbf{U}$, satisfying

$$\mathbf{U}\mathbf{U}^\dagger = \mathbf{U}^\dagger\mathbf{U} = 1. \tag{65}$$

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

---

[14] 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 these notes the term "state" can always be taken to mean "pure state" unless explicitly noted otherwise.

[15] 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".

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

the action of **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. Therefore it takes unit vectors (defined by the sums of the moduli of the squared amplitudes being unity) into unit vectors and is indeed unitary. Many (but by no means all) of the unitary operations on Qbits we shall examine 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 these trivial 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*), 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. (In a reversible classical computer it turns out that one needs 3-Cbit gates to build up general logical operations. One of the many charms of quantum computation is that 2-Qbit gates suffice for this purpose, as we shall see in Chapter II.)

E. Measurement of Qbits.

To specify the state of a single Cbit you need only one bit of information: is the state of the Cbit $|0\rangle$ or $|1\rangle$?. But to specify the state 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 subject only to the normalization constraint (58). 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 transformation, 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 has a value which is either 0 or 1, and you can find out which value each has just by looking. There is nothing problematic about learning the state of the Cbits, and hence learning the result of any calculation you may have built up out of operations on them. Nor is the act of acquiring this information disruptive: the state of Cbits is unaltered by the process of reading them.

In stark contrast, if you have n Qbits in some superposition of computational basis states, there is nothing you can do to to them to learn the values of the amplitudes in that superposition. You cannot read out those values and you cannot find out what the state is. There is nothing inherent in the Qbits that you can get at to learn the state. There is no way to extract the vast amount of information contained in the amplitudes α_x. There is only one thing you can do to extract information from n Qbits in a given state: it is called *making a measurement*.[17]

[17] Physicists will note that what follows is more accurately characterized as "making

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 the test produces is in general not 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

$$|\psi\rangle_n = \sum_{0 \le x < 2^n} \alpha_x |x\rangle_n, \tag{66}$$

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

$$p_\psi(x) = |\alpha_x|^2. \tag{67}$$

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 measure — i.e. test — the Qbits. The squared magnitudes of the amplitudes give the probabilities of outcomes of measurements. Normalization conditions like (60) are just the requirements that the probabilities for all of the 2^n mutually exclusive outcomes add up to 1.

The simplest statement of the Born rule is for a single Qbit. If the state of the Qbit is the superposition (57) 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 $|\alpha_1|^2$. One can think of the measurement as being performed by a special kind of gate — a *measurement gate*, which, acting on a Qbit whose state is $\alpha_0|0\rangle + \alpha_1|1\rangle$, indicates a 0 or 1 on a digital display, with probabilities $|\alpha_0|^2$ and $|\alpha_1|^2$. A measurement gate for n Qbits in the state (61) indicates x with probability $|\alpha_x|^2$. It turns out, as a consequence of the "generalized Born rule" described below, that one can construct a measurement gate for n-Qbits out of n 1-Qbit measurement gates, subjecting each of the Qbits to its own 1-Qbit gate.

You might think that by subjecting a given set of n Qbits to a measurement over and over again, you could accumulate enough statistics to get a good estimate of at least the magnitudes (though not the phases) of all the amplitudes α_x. But this doesn't work at

---

a measurement in the computational (classical) basis." There are other ways to make 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, unless otherwises noted. Because unitary transformations can be applied prior to such measurements no generality is lost by this strictly terminological convention.

all, because of another irritating feature of Qbits. In contrast to Cbits, once you make just a single measurement of each Qbit, you remove the possibility of extracting any further information about their original state $|\psi\rangle$. After such a measurement, if the result is, for example, 01100, then the post-measurement state of 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. The Qbits that emerge from a measurement gate that indicates the outcome x are characterized by the state $|x\rangle$. 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.[18]

You might wonder how one can learn anything at all of computational interest under such 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 very close to zero, with useful information being carried by any of the values of x that have a significant probability of being indicated by the measurement. It is also 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 the occasional irrelevant low probability outcome. How this is done in various cases of interest will be one of our major preoccupations.

An important generalization of the Born rule comes into play when not all of the Qbits are measured. (This extended Born rule, though of great importance in quantum computation, is not always mentioned in conventional expositions of quantum mechanics.) Suppose you have $m+n$ Qbits in a state $|\psi\rangle_{m+n}$ and you measure only m of them. The expansion of $|\psi\rangle_{m+n}$ in the computational basis can be written as

$$|\psi\rangle_{m+n} = \sum_{xy} \alpha_{xy} |x\rangle_m |y\rangle_n, \tag{68}$$

where I have written the m Qbits to be measured on the left. This state is of the general form

$$|\psi\rangle_{m+n} = \sum_{x} \alpha_{x} |x\rangle_m |\phi_x\rangle_n \tag{69}$$

where $\sum_x |\alpha_x|^2 = 1$ and the states $|\phi_x\rangle_n$ are normalized, but not necessarily orthogonal. The form (69) does indeed reduce to (68) if the normalized states $|\phi_x\rangle$ are given by

$$|\phi_x\rangle_n = \alpha_x^{-1} \sum_{y} \alpha_{xy} |y\rangle_n, \tag{70}$$

[18] This should not be taken to imply (though, alas, it often is) that the Qbits themselves suffer a catastrophic "reduction" or "collapse", since the state is not an internal property of the Qbits.

with amplitudes α_x given by

$$\alpha_x = \sqrt{\sum_y |\alpha_{xy}|^2}. \tag{71}$$

The generalized Born rule asserts that if only the m Qbits on the left are measured, then one will find the result x with probability $|\alpha_x|^2$ and after the measurement the state of all $m+n$ Qbits, will be the unentangled product state

$$|x\rangle_m |\phi_x\rangle_n \tag{72}$$

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

You should convince yourself that this extension of the ordinary Born rule, which can be viewed as the $n = 0$ case of the more general rule, satisfies an essential consistency condition. Immediately after measuring the first m Qbits and getting the value x, one can measure the remaining n of them. The statistical distribution of combined outcomes that results from applying the ordinary Born rule to the measurement of the remaining n Qbits, now in the state $|\phi_x\rangle$, is exactly the same as that given by applying the ordinary Born rule to a joint measurement of all $m+n$ Qbits.

It is extremely important immediately to note and 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 revealed by the test, since, for example, the action of the test itself might well play a role in bringing forth the outcome.

Indeed, it is easy to 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 using the Hadamrd transformation (37). Although 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 $|1\rangle$, we can extend their action to arbitrary linear combination of these states by requiring them to be linear operators. Since the states $|0\rangle$ and $|1\rangle$ form a basis, this determines the action of $\mathbf{X}$ and $\mathbf{Z}$ on any 1-Qbit state.

Because it is hermitian and its own inverse, $\mathbf{H}$ is unitary, and therefore the kind of operation a quantum computer can apply to the state of a Qbit. The result of the operation is to change the state $|\phi\rangle$ of the Qbit to $\mathbf{H}|\phi\rangle$. If we apply $\mathbf{H}$ to a Qbit in the state

$$|\phi\rangle = \tfrac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \tag{73}$$

it follows from (41) that the result is just

$$\mathbf{H}|\phi\rangle = |0\rangle. \tag{74}$$

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

But suppose a Qbit in the state $|\phi\rangle$ were actually 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 (41), the subsequent action of $\mathbf{H}$ would produce a state that under measurement yielded 0 or 1 with equal probability. This contradicts the fact which we have extracted directly from (74), that the result of making a measurement on a Qbit in the state $\mathbf{H}|\phi\rangle$ is invariably 0.

Therefore a Qbit in a quantum superposition of $|0\rangle$ and $|1\rangle$ definitely cannot be regarded as being either in the state $|0\rangle$ or in the state $|1\rangle$ with certain probabilities. Its state is something altogether different from either of these. 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 will not in general be either $|0\rangle$ or $|1\rangle$, but a superposition of the form (57). Such a superposition is no more or less natural and irreducible a description of a Qbit than $|0\rangle$ and $|1\rangle$ are. (See Section A2 of the Appendix to this Chapter.)

Note that if the state (66) 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 (67) 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 reduces to 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 calculation, measurement gates also play a role (not usually 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 have no state of its own at all — then how can one produce a set of Qbits in a definite state for the gates of a quantum computer to act on? The answer is by measurement, and only 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.

F. Further Features of Dirac Notation.

The features of Dirac notation developed above are the only ones needed to understand

the physical behavior of Qbits and to read much of what follows. There are, however, several mathematical embellishments of the notation which are quite useful, and which you will certainly encounter if you delve into the literature on quantum computation and quantum information.

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). The inner product is a number satisfying the usual rules for inner products in a vector space with complex scalars:

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

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

$$\langle\phi|\phi\rangle > 0, \quad |\phi\rangle \neq 0. \tag{75}$$

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

$$\langle 0|0\rangle = \langle 1|1\rangle = 1, \quad \langle 1|0\rangle = \langle 0|1\rangle = 0. \tag{76}$$

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

$$\langle x|y\rangle = 0, \ x \neq y; \quad = 1, \ x = y, \tag{77}$$

since the inner product of two such n-fold tensor products 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

$$\big(|\phi\rangle\big)^\dagger \text{ or } \langle\phi|, \tag{78}$$

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 $|\psi\rangle$:

$$\langle\phi|\psi\rangle = \langle\phi|\big(|\psi\rangle\big). \tag{79}$$

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 (79) encloses the names of the two vectors between the brackets $\langle \ \rangle$.

The duals $\big(|x\rangle\big)^\dagger = \langle x|$ of the computational basis vectors are defined by linearity and their action (77) on the computational basis. To make the rules (75) for the inner product come out right we must have

$$\big(\alpha|\psi\rangle + \beta|\phi\rangle\big)^\dagger = \alpha^*\langle\psi| + \beta^*\langle\phi|, \tag{80}$$

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

$$|\psi\rangle = \sum_{x=0}^{2^{n-1}} \alpha_x |x\rangle_n \tag{81}$$

by the column vector

$$\begin{pmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_2 \\ \vdots \end{pmatrix} \tag{82}$$

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

$$\big(\alpha_0^*\ \alpha_1^*\ \alpha_2^* \ldots\big), \tag{83}$$

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

$$|\phi\rangle = \sum_{x=0}^{2^{n-1}} \beta_x |x\rangle_n \tag{84}$$

is just the ordinary matrix product:

$$\langle\phi|\psi\rangle = \big(\beta_0^*\ \beta_1^*\ \beta_2^* \ldots\big) \begin{pmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_2 \\ \vdots \end{pmatrix} = \sum \beta_x^* \alpha_x. \tag{85}$$

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

$$\langle\psi|\mathbf{A} = \big(\mathbf{A}^\dagger|\psi\rangle\big)^\dagger. \tag{86}$$

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."

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

$$\langle\psi|\mathbf{A}|\phi\rangle^* = \langle\psi|\mathbf{A}|\phi\rangle^\dagger = \langle\phi|\mathbf{A}^\dagger|\psi\rangle, \tag{87}$$

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

$$\big(|\psi\rangle\langle\phi|\big)|\chi\rangle = |\psi\rangle\big(\langle\phi|\chi\rangle\big). \tag{88}$$

If $|\psi\rangle$ and $|\phi\rangle$ have expansions (81) and (84) then the matrix expressing their outer product in the computational basis is given by the ordinary matrix product

$$\begin{pmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_2 \\ \vdots \end{pmatrix} \big(\beta_0^* \ \beta_1^* \ \beta_2^* \ldots\big), \tag{89}$$

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

$$M_{xy} = \langle x|\big(|\psi\rangle\langle\phi|\big)|y\rangle \tag{90}$$

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

$$\big(\langle x|\psi\rangle\big)\big(\langle\phi|y\rangle\big). \tag{91}$$

The latter form (and the expansions (81) and (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

$$\mathbf{P}_\psi = |\psi\rangle\langle\psi| \tag{92}$$

(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:

$$\mathbf{1} = \sum_{i=1}^{n} |\psi_i\rangle\langle\psi_i|. \tag{93}$$

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

$$|\phi\rangle = \sum_{i=1}^{n} |\psi_i\rangle\langle\psi_i|\phi\rangle. \tag{94}$$

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:

$$\mathbf{A} = \mathbf{1A1} = \sum_{i,j=1}^{n} |\psi_i\rangle\langle\psi_i|A|\psi_j\rangle\langle\psi_j| = \sum_{i,j=1}^{n} \big(\langle\psi_i|A|\psi_j\rangle\big)|\psi_i\rangle\langle\psi_j|\,. \tag{95}$$

The Born rule, relating the amplitudes in the expansion (57) 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

$$p_\psi(0) = |\langle 0|\psi\rangle|^2, \quad p_\psi(1) = |\langle 1|\psi\rangle|^2. \tag{96}$$

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

$$p_\psi(0) = \langle 0|\mathbf{P}_\psi|0\rangle, \quad p_\psi(1) = \langle 1|\mathbf{P}_\psi|1\rangle. \tag{97}$$

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

$$p_\psi(0) = \langle\psi|\mathbf{P}_0|\psi\rangle, \quad p_\psi(1) = \langle\psi|\mathbf{P}_1|\psi\rangle. \tag{98}$$

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

$$p_\Psi(x) = |\langle x|\Psi\rangle|^2 = \langle x|\mathbf{P}_\Psi|x\rangle = \langle\Psi|\mathbf{P}_x|\Psi\rangle, \tag{99}$$

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

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.

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 devopments of a slightly more technical nature, that we will make only occasional use of. I recommend looking at them now and then examining them more closely when their contents are needed. Section A3 applies some of this formalism to illustrate one of the most striking peculiarities of Qbits.

A1. 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 can occasionally be useful in applications to quantum computation.

The unit operator $\mathbf{1}$ and the three Pauli matrices (48) form a basis for the four dimensional algebra of two dimensional matrices, so any two-dimensional matrix has a unique expansion of the form[19,20]

$$\mathbf{u} = u_0 + \mathbf{u} \cdot \sigma \tag{100}$$

for some complex number u_0 and 3-vector $\mathbf{u}$ with complex components u_x, u_y, and u_z. Here σ signifies the "3-vector" whose components are the Pauli matrices σ_x, σ_y, and σ_z given in (48), so $\mathbf{u} \cdot \sigma = u_x\sigma_x + u_y\sigma_y + u_z\sigma_z$.

Let us now impose on (100) the condition (64) 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

$$\mathbf{u}^\dagger = u_0 + \mathbf{u}^* \cdot \sigma. \tag{101}$$

[19] 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.

[20] 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 more cumbersome, 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 rotation matrices $\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$ and $\begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$.

The rule (55) then tells us that for $\mathbf{u}$ to be unitary we must have

$$0 = 1 - \mathbf{u}^\dagger\mathbf{u} = 1 - u_0^2 - \mathbf{u}^* \cdot \mathbf{u} - \big(u_0(\mathbf{u} + \mathbf{u}^*) + i\mathbf{u}^* \times \mathbf{u}\big) \cdot \sigma, \tag{102}$$

Since $\mathbf{1}$ and the three Pauli matrices are linearly independent in the 4-dimensional algebra of 1-Qbit operators, the coefficients of all four in (102) must vanish and we have

$$1 = u_0^2 + \mathbf{u}^* \cdot \mathbf{u}, \quad 0 = u_0(\mathbf{u} + \mathbf{u}^*) + i\mathbf{u}^* \times \mathbf{u}. \tag{103}$$

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

$$u_0 \mathrm{Re}\,\mathbf{u} = \mathrm{Re}\,\mathbf{u} \times \mathrm{Im}\,\mathbf{u}. \tag{104}$$

If $u_0 \neq 0$, it follows that Re$\mathbf{u}$·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 (104) 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,

$$\mathbf{u} = u_0 + i\mathbf{v} \cdot \sigma, \tag{105}$$

where u_0 is a real number, $\mathbf{v}$ is a real vector, and, from the first of (103),

$$u_0^2 + \mathbf{v} \cdot \mathbf{v} = 1. \tag{106}$$

The identity (106) is ensured by parametrizing u_0 and $\mathbf{v}$ in terms of a real unit vector[21] $\mathbf{n}$ parallel to $\mathbf{v}$, and a real angle γ so that

$$\mathbf{u} = \cos\gamma + i(\mathbf{n} \cdot \sigma)\sin\gamma. \tag{107}$$

An alternative way of writing (107) is

$$\mathbf{u} = e^{i\gamma\mathbf{n}\cdot\sigma}. \tag{108}$$

(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 (55). The argument is exactly the same as that establishing that $e^{i\varphi} = \cos\varphi + i\sin\varphi$ for any real number φ.)

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

A connection between these 2-dimensional unitary matrices and ordinary three dimensional rotations emerges from the fact that each of the three Pauli matrices (48) has zero trace,[22] and the fact that the operator unitary transformation

$$\mathbf{A} \to \mathbf{u}\mathbf{A}\mathbf{u}^\dagger \tag{109}$$

preserves the trace of $\mathbf{A}$.

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 $\mathbf{a}\cdot\sigma$ has zero trace 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:

$$\mathbf{u}(\mathbf{a}\cdot\sigma)\mathbf{u}^\dagger = \overline{\mathbf{a}}\cdot\sigma. \tag{110}$$

It follows that

$$\mathbf{u}(\mathbf{a}\cdot\sigma)(\mathbf{b}\cdot\sigma)\mathbf{u}^\dagger = \big(\mathbf{u}(\mathbf{a}\cdot\sigma)\mathbf{u}^\dagger\big)\big(\mathbf{u}(\mathbf{b}\cdot\sigma)\mathbf{u}^\dagger\big) = \big(\overline{\mathbf{a}}\cdot\sigma\big)\big(\overline{\mathbf{b}}\cdot\sigma\big). \tag{111}$$

Since unitary transformations preserve the trace,

$$\mathrm{Tr}(\mathbf{a}\cdot\sigma)(\mathbf{b}\cdot\sigma) = \mathrm{Tr}(\overline{\mathbf{a}}\cdot\sigma)\big(\overline{\mathbf{b}}\cdot\sigma\big). \tag{112}$$

Hence, from (55),

$$\overline{\mathbf{a}}\cdot\overline{\mathbf{b}} = \mathbf{a}\cdot\mathbf{b}. \tag{113}$$

But the most general real, linear,[23] inner-product-preserving transformation on real 3-vectors is a rotation. Consequently the transformation from $\mathbf{a}$ to $\overline{\mathbf{a}}$ induced by $\mathbf{u}$ through (110) is a rotation:

$$\overline{\mathbf{a}} = \mathbf{R}_\mathbf{u}\,\mathbf{a}. \tag{114}$$

By applying the product $\mathbf{uv}$ of two unitary transformations in two steps,

$$(\mathbf{uv})(\mathbf{a}\cdot\sigma)(\mathbf{uv})^\dagger = \mathbf{u}\big(\mathbf{v}(\mathbf{a}\cdot\sigma)\mathbf{v}^\dagger\big)\mathbf{u}^\dagger = \mathbf{u}\big(R_\mathbf{v}\mathbf{a}\cdot\sigma\big)\mathbf{u}^\dagger = \mathbf{R}_\mathbf{u}\mathbf{R}_\mathbf{v}\mathbf{a}\cdot\sigma, \tag{115}$$

we deduce that

$$\mathbf{R}_\mathbf{uv} = \mathbf{R}_\mathbf{u}\mathbf{R}_\mathbf{v}. \tag{116}$$

Thus the association of three-dimensional rotations with two-dimensional unitary matrices preserves the multiplicative structure of the rotation group.

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

[23] It follows directly from (110) that $\overline{\mathbf{a}+\mathbf{b}} = \overline{\mathbf{a}} + \overline{\mathbf{b}}$ and $\overline{\lambda\mathbf{a}} = \lambda\overline{\mathbf{a}}$.

When the vector $\mathbf{a}$ in (110) is taken to be the vector $\mathbf{n}$ appearing in $\mathbf{u}$ (in (107) or (108)) 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

$$\cos\theta = \mathbf{m}\cdot\overline{\mathbf{m}}. \tag{117}$$

We then have

$$\begin{aligned}\cos\theta &= \tfrac{1}{2}\mathrm{Tr}\big((\mathbf{m}\cdot\sigma)(\overline{\mathbf{m}}\cdot\sigma)\big)\\ &= \tfrac{1}{2}\mathrm{Tr}\big((\mathbf{m}\cdot\sigma)(\cos\gamma + i\sin\gamma\ \mathbf{n}\cdot\sigma)(\mathbf{m}\cdot\sigma)(\cos\gamma - i\sin\gamma\ \mathbf{n}\cdot\sigma)\big)\\ &= \tfrac{1}{2}\mathrm{Tr}\big((\cos\gamma\mathbf{m} - \sin\gamma\mathbf{m}\times\mathbf{n})\cdot\sigma)(\cos\gamma\mathbf{m} + \sin\gamma\mathbf{m}\times\mathbf{n})\cdot\sigma)\big)\\ &= \cos^2\gamma - \sin^2\gamma = \cos(2\gamma),\end{aligned} \tag{118}$$

where we have made repeated use of (55) and the fact that $\mathbf{m}\cdot\mathbf{n} = 0$. So the unitary matrix (108) or (107) 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.[24] 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:

$$\mathbf{u}(\mathbf{n},\theta) = e^{i(\theta/2)(\mathbf{n}\cdot\sigma)} = \cos\tfrac{1}{2}\theta + i(\mathbf{n}\cdot\sigma)\sin\tfrac{1}{2}\theta. \tag{119}$$

If you have never seen this before, note that it actually reduces some nontrivial three-dimensional geometry to simple algebra, just as Euler's relation $e^{i\phi} = \cos\phi + i\sin\phi$ reduces some nontrivial two-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 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.

$$\cos\tfrac{1}{2}\gamma + i(\mathbf{c}\cdot\sigma)\sin\tfrac{1}{2}\gamma = \big(\cos\tfrac{1}{2}\alpha + i(\mathbf{a}\cdot\sigma)\sin\tfrac{1}{2}\alpha\big)\big(\cos\tfrac{1}{2}\beta + i(\mathbf{b}\cdot\sigma)\sin\tfrac{1}{2}\beta\big), \tag{120}$$

and expand the right side using (55). To get the angle take the trace of both sides (or identify the coefficients of unity) to find

$$\cos\tfrac{1}{2}\gamma = \cos\tfrac{1}{2}\alpha\cos\tfrac{1}{2}\beta - (\mathbf{a}\cdot\mathbf{b})\sin\tfrac{1}{2}\alpha\sin\tfrac{1}{2}\beta. \tag{121}$$

[24] The rotations 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 (105) 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.

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

$$\sin\tfrac{1}{2}\gamma\ \mathbf{c} = \sin\tfrac{1}{2}\beta\cos\tfrac{1}{2}\alpha\ \mathbf{b} + \sin\tfrac{1}{2}\alpha\cos\tfrac{1}{2}\beta\ \mathbf{a} - \sin\tfrac{1}{2}\alpha\sin\tfrac{1}{2}\beta\,(\mathbf{a}\times\mathbf{b}). \tag{122}$$

A2. 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 $\mathbf{u}$ taking the computational basis states $|0\rangle$ and $|1\rangle$ into $|\phi\rangle$ and $|\psi\rangle$.[25] Now the computational basis states are eigenstates of the 1-Qbit number operator:

$$\mathbf{n}|x\rangle = x|x\rangle, \quad x = 0 \text{ or } 1, \tag{123}$$

where

$$\mathbf{n} = \tfrac{1}{2}(1-\sigma_z) = \tfrac{1}{2}(1-\mathbf{z}\cdot\sigma). \tag{124}$$

Since

$$|\phi\rangle = \mathbf{u}|0\rangle, \quad |\psi\rangle = \mathbf{u}|1\rangle, \tag{125}$$

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

$$\overline{\mathbf{n}}|\phi\rangle = 0, \quad \overline{\mathbf{n}}|\psi\rangle = |\psi\rangle. \tag{126}$$

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

$$\overline{\mathbf{n}} = \mathbf{u}\,\mathbf{n}\,\mathbf{u}^\dagger = \tfrac{1}{2}\big(1-\mathbf{u}(\mathbf{z}\cdot\sigma)\mathbf{u}^\dagger\big) = \tfrac{1}{2}(1-\overline{\mathbf{z}}\cdot\sigma), \tag{127}$$

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

Thus $\overline{\mathbf{n}}$, which functions as a number operator for the states $|\phi\rangle = \mathbf{u}(\mathbf{m},\theta)|0\rangle = |\overline{0}\rangle$ and $|\psi\rangle = \mathbf{u}(\mathbf{m},\theta)|1\rangle = |\overline{1}\rangle$ is constructed out of the component of the vector of operators σ along the direction $\overline{\mathbf{z}} = \mathbf{R}(\mathbf{m},\theta)\mathbf{z}$ 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, $|\overline{0}\rangle = \mathbf{u}|0\rangle$ and $|\overline{1}\rangle = \mathbf{u}|1\rangle$ 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.

[25] 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.

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 **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 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 Qbit (which is not always that easy to arrange) can be replaced by straightforwardly applying the corresponding rotation to the apparatus that measures the Qbit.

A3. An application of the formalism: "Spooky action at a distance"

We examine here the entangled 2-Qbit state[26]

$$|\Psi\rangle = \tfrac{1}{\sqrt{3}}\big(|00\rangle + |01\rangle + |10\rangle\big) \tag{128}$$

which can also be written in terms of the Hadamard operator (41) as

$$|\Psi\rangle = \tfrac{2}{\sqrt{3}}\big(\mathbf{H}_a\mathbf{H}_b|00\rangle - \tfrac{1}{2}|11\rangle\big), \tag{129}$$

where we take the 1-Qbit operators $\mathbf{H}_a$ and $\mathbf{H}_b$ to act on the left and right Qbits respectively (and as the identity on the other Qbit). Suppose Alice has possession of the left Qbit and Bob has possession of the right Qbit, and suppose the Qbits are in the 2-Qbit state $|\Psi\rangle$. Alice and Bob are each going to measure their Qbit. Each is given the opportunity to apply (or not apply) a Hadamard transformation to their own Qbit before making their measurement, choices they each make independently by tossing their own coin. So prior to their measurements, the state Ψ can be further modified by one of four unitary transformations:

$$\text{Case 1} : \mathbf{1}; \quad \text{Case 2} : \mathbf{H}_a; \quad \text{Case 3} : \mathbf{H}_b; \quad \text{or} \quad \text{Case 4} : \mathbf{H}_a\mathbf{H}_b. \tag{130}$$

[26] The strange properties of a family of states, of which this is the simplest example, were first pointed out about ten years ago by Lucien Hardy.

In Case 1 Ψ is unmodified. It is evident from the form (128) that $\langle 11|\Psi\rangle = 0$ so Alice and Bob cannot both find the value 1 when they each measure their Qbit. The probability of that happening is 0.

In Case 2, since $\mathbf{H}^2 = 1$, prior to the measurements the state is transformed to

$$\mathbf{H}_a|\Psi\rangle = \tfrac{2}{\sqrt{3}}\big(\mathbf{H}_b|00\rangle - \tfrac{1}{2}\mathbf{H}_a|11\rangle\big). \tag{131}$$

This is orthogonal to[27] $|10\rangle$ so when the Qbits are measured Bob cannot find the value 0 if Alice finds the value 1.

In Case 3 (which differs from Case 2 only by the interchange of Alice and Bob) when the Qbits are measured Alice cannot find the value 0 if Bob finds the value 1.

In Case 4, prior to the measurements the state is transformed to

$$\mathbf{H}_a\mathbf{H}_b|\Psi\rangle = \tfrac{2}{\sqrt{3}}\big(|00\rangle - \tfrac{1}{2}\mathbf{H}_a\mathbf{H}_b|11\rangle\big), \tag{132}$$

so the probability of Alice and Bob both finding the value 1 is

$$|\langle 11|\mathbf{H}_a\mathbf{H}_b|\Psi\rangle|^2 = \tfrac{1}{3}|\langle 11|\mathbf{H}_a\mathbf{H}_b|11\rangle|^2 \tag{133}$$

which is nonzero, and easily shown from the definition (41) of $\mathbf{H}$ to be equal to $\frac{1}{12}$.

Now notice something rather strange about these statistics. Entertain the plausible hypothesis that the value Bob finds upon measuring his Qbit does not depend on whether or not Alice applied a Hadamard transform to her own Qbit, and vice-versa. The hypothesis is plausible, because Alice and Bob and their Qbits can be so very far apart that it is highly unlikely that anything done by Alice could affect what happens in the vicinity of Bob, and vice-versa. But now consider the implications of this hypothesis for the situation in which both apply Hadamard transformations to their Qbits (Case 4) and both subsequent measurements give the value 1. The probability of this happening is small — 1/48 — but not zero. Since it can happen, occasionally it will happen.

According to our hypothesis Bob would have got the value 1 even if Alice had not applied a Hadamard to her Qbit. But if Bob applied a Hadamard and Alice did not then we are in Case 3, and in Case 3 if Bob gets 1 Alice cannot get 0. She must get 1. But our hypothesis also specifies that what Alice gets does not depend on whether or not Bob applied a Hadmard transform to his own Qbit, and therefore if Alice had not applied a Hadamard transform to her Qbit, she would have got 1 whether or not Bob applied a Hadamard to his.

By the same reasoning (interchanging Alice and Bob — the situation is completely symmetric) we conclude that if Bob had not applied a Hadamard transformation to his Qbit, he would have got 1 whether or not Alice applied a Hadamard to hers.

[27] If this is not evident, expand $\mathbf{H}_b|00\rangle$ and $\mathbf{H}_a|11\rangle$ to the forms $|0\rangle\otimes\big(\mathbf{H}|0\rangle\big)$ and $\big(\mathbf{H}|1\rangle\big)\otimes|1\rangle$, both of which are orthogonal to $|10\rangle = |1\rangle\otimes|0\rangle$.

So if both apply Hadamards and both get 1 when they measure their Qbit, our hypothesis leads to the conclusion that if neither had applied Hadamards both would still have got 1.

But neither of them applying a Hadamard is Case 1, and in Case 1 both never get 1!

The hypothesis that the value one of them finds upon measuring does not depend on whether the other applied a Hadamard before measuring leads to a contradiction. Over the years[28] 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 her or his own Qbit before measuring. They call this "quantum nonlocality" or "spooky action at a distance"[29].

My own take on it is rather different. The hypothesis that led us to a contradiction is not so much wrong, as it is devoid of meaning. One of the things quantum mechanics teaches us is that "does not depend on" is an extremely dangerous phrase to apply to the outcome of a single measurement. With any given pair of Qbits, Alice and Bob each either does or does not apply a Hadamard prior to their measurement. Only one of the four 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 entails unique subsequent behavior. But in the intrinsically nondeterminstic world that we actually inhabit, it makes no sense to insist that Bob "would have got" the same result if we turned back the clock and allowed Alice not to perform the Hadamard that she actually did perform. For Bob to have to get the same result in this fictitious world requires more than just the irrelevance of Alice's decision whether or not to perform the Hadamard. It also requires that whatever it is that actually is relevant to Bob's outcome remains the same in both worlds *and* is capable of completely determining the outcome. But in the quantum world there is an irreducible randomness to such outcomes: nothing is capable of predetermining them.[30]

It is, however, meaningful to ask whether the *statistics* of the values Bob finds upon measuring his Qbit depend on whether or not Alice applied a Hadamard transform to her Qbit, or vice versa, since one can imagine Alice and Bob repeatedly playing this game, with many pairs of Qbits, always starting with the same initial 2-Qbit state (128). In that case they could 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

[28] Although Hardy only discovered this particular argument ten years ago, closely related situations have been known since a famous paper by John Bell appeared in 1964.

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

[30] 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.

apply a Hadamard to her faraway Qbit before she measured it, this would be *non*spooky action at a distance which could actually be used to send messages. It is important to verify 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 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

$$p_\Psi(x, y) = \langle\Psi|\mathbf{P}^a_x\mathbf{P}^b_y|\Psi\rangle, \tag{134}$$

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

$$|\Phi\rangle = \mathbf{U}_a\mathbf{U}_b|\Psi\rangle. \tag{135}$$

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

$$\begin{aligned} p_\Psi(x, y|\mathbf{U}_a, \mathbf{U}_b) &= \langle\Phi|\mathbf{P}^a_x\mathbf{P}^b_y|\Phi\rangle = \langle\Psi|\mathbf{U}^\dagger_b\mathbf{U}^\dagger_a\big(\mathbf{P}^a_x\mathbf{P}^b_y\big)\mathbf{U}_a\mathbf{U}_b|\Psi\rangle \\ &= \langle\Psi|(\mathbf{U}^\dagger_a\mathbf{P}^a_x\mathbf{U}_a)(\mathbf{U}^\dagger_b\mathbf{P}^b_y\mathbf{U}_b)|\Psi\rangle, \end{aligned} \tag{136}$$

(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). Since

$$\sum_x \mathbf{U}^\dagger_a\mathbf{P}^a_x\mathbf{U}_a = \mathbf{U}^\dagger_a\Big(\sum_x \mathbf{P}^a_x\Big)\mathbf{U}_a = \mathbf{U}^\dagger_a\mathbf{1}\mathbf{U}_a = 1, \tag{137}$$

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

$$\sum_x p_\Psi(x, y|\mathbf{U}_a, \mathbf{U}_b) = \langle\Psi|(\mathbf{U}^\dagger_b\mathbf{P}^b_y\mathbf{U}_b)|\Psi\rangle = p_\Psi(y|\mathbf{U}_b). \tag{138}$$

So 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 [for example by the application of appropriate 2-Qbit gates] with those in the original group).