Last revised 3/30/06

LECTURE NOTES ON QUANTUM COMPUTATION

Cornell University, Physics 481-681, CS 483; Spring, 2005

IV. Searching with a Quantum Computer

Suppose you know that exactly one n-bit integer satisfies a certain condition, and suppose you have a black-boxed subroutine that acts on the $N = 2^n$ different n-bit integers, outputting 1 ("true") if the integer is the special one and 0 ("false") otherwise. In the absence of any other information, to find the special integer you can do no better with a classical computer than to apply repeatedly the subroutine to different random numbers until you hit on the special one. If you apply it to M different integers the probability of your finding the special number is M/N. You must test $\frac{1}{2}N$ different integers to have a 50% chance of success.

If, however, you have a quantum computer with a subroutine that performs such a test, then for large (but not astronomically large) N, you can find the special integer with a probability that is very close to 1, by a method that requires you to call the subroutine only $(\pi/4)\sqrt{N}$ times. This very general capability of quantum computers was discovered by Lov Grover, and goes under the name of *Grover's search algorithm*. Shor's period-finding algorithm, Grover's search algorithm, along with their various modifications and extensions, constitute the two masterpieces of quantum computational software. One can think of the black-boxed subroutine in various ways. It could perform a mathematical calculation to determine whether the input integer is the special one. Here is an example:

If an odd number p can be expressed as the sum of two squares, $m^2 + n^2$, then since one of m or n must be even and the other odd, p must be of the form $4k + 1$. It is a fairly elementary theorem of number theory that if p is a *prime* number of the form $4k+1$ then it can always be expressed as the sum of two squares in exactly one way. (Thus 5 = 4+1, 13 = 9+4, 17 = 16+1, 29 = 25 + 4, 37 = 36+1, 41 = 25+16, etc.) Given any such prime p, the simple-minded way to find the two squares is to take randomly selected integers x with $1 \le x \le N$, with N the largest integer less than $\sqrt{p/2}$, until you find the one for which $\sqrt{p - x^2}$ is an integer a. If p is of the order of a trillion, then following the simple-minded procedure you would have to calculate $\sqrt{p - x^2}$ for nearly a million x to have a better than even chance of succeeding. But using Grover's procedure with an appropriately programmed quantum computer you could succeed with a probability of success extremely close to 1 with by calling the quantum subroutine that evaluated $\sqrt{p - x^2}$ fewer than a thousand times.

Mathematically well-informed friends tell me that this particular example admits of much more efficient ways to proceed than random testing, but the quantum algorithm to be

described below enables even mathematical ignoramuses like me, equipped with a quantum computer, to do better than random testing by a factor of $1/\sqrt{N}$. And Grover's algorithm will provide this speed-up even on problems that might stump the mathematically well-informed.

Alternatively, the black box could contain Qbits that have been loaded with a body of data — for example alphabetically ordered names and phone numbers — and one might be looking for the name that went with a particular phone number. It is with this kind of application in mind that the Grover's neat trick has been called searching a database. Using as precious a resource as Qbits, however, merely to store classical information would be insanely extravagant, given our current or currently forseeable ability to manufacture Qbits. Finding a unique solution — or one of a small number of solutions (see Section C below) — to a tough mathematical puzzle seems a more promising application.

A. The Grover iteration.

Grover's algorithm requires a quantum subroutine that indicates, when presented with any n-bit integer, whether or not that integer is the special one a being sought, returning this information as the value of a function $f(x)$ that satisfies

$$f(x) = 0,\ x \neq a; \qquad f(x) = 1,\ x = a. \tag{4.1}$$

Grover discovered a completely general way to do significantly better than merely letting the subroutine operate on different numbers from a list of 2^n candidates, until it produced the output 1 — the best classical method. The quantum computational speed-up relies on the usual implementation of the subroutine that calculates f, in the form of a unitary transformation $\mathbf{U}_f$ that acts on an n-Qbit input register that contains x and a 1-Qbit output register that is or is not flipped from 0 to 1, depending on whether x is or is not the number a having the special property:

$$\mathbf{U}_f\Big(|x\rangle_n|y\rangle_1\Big) = |x\rangle_n|y \oplus f(x)\rangle_1. \tag{4.2}$$

A simple circuit that has precisely this action is shown in Figure 4.1. The figure can be viewed as providing a minimalist version of Grover's algorithm, reminiscent of the Bernstein-Vazirani problem (Chapter 2), though not susceptible to the special trick that worked in that simpler case. We are given a black box containing a circuit of the type shown in Figure 4.1, but we are not told which Qbits are acted on by the $\mathbf{X}$ gates and which are not — information specified by the unknown n-bit integer a. If there were n Qbits in the input register and the computer were classical, we could do no better than to try each of the 2^n possible inputs until we found the one for which the output register was flipped. But using Grover's algorithm we can determine this information with probability quite close to 1, by invoking the subroutine fewer than $2^{n/2}$ times — more precisely $(\pi/4)2^{n/2}$) times — when N is large.

As in the Bernstein-Vazarani problem, it is useful to change the flip of the state of the output register into an overall sign change, by transforming the 1-Qbit output register into the state

$$\mathbf{H}|1\rangle = \tfrac{1}{\sqrt{2}}\big(|0\rangle - |1\rangle\big) \tag{4.3}$$

prior to the application of $\mathbf{U}_f$. The action of $\mathbf{U}_f$ is then to multiply the $(n+1)$-Qbit state by -1 if and only if $x = a$:

$$\mathbf{U}_f\Big(|x\rangle \otimes \mathbf{H}|1\rangle\Big) = (-1)^{f(x)}|x\rangle \otimes \mathbf{H}|1\rangle. \tag{4.4}$$

In this form, the effect of $\mathbf{U}_f$ on the states $|x\rangle \otimes \mathbf{H}|1\rangle$, is exactly the same as doing nothing at all to the 1-Qbit output register, while acting on the n-Qbit input register with an n-Qbit unitary transformation $\mathbf{V}$ that acts on the computational basis as follows:

$$\mathbf{V}|x\rangle = (-1)^{f(x)}|x\rangle = \begin{cases} |x\rangle, & x \neq a, \\ -|a\rangle, & x = a. \end{cases} \tag{4.5}$$

We can write $\mathbf{V}$ as

$$\mathbf{V} = \mathbf{1} - 2|a\rangle\langle a|, \tag{4.6}$$

where $|a\rangle\langle a|$ is the projection operator[1] on the state $|a\rangle$. Since $\mathbf{U}_f$ is linear, so is $\mathbf{V}$. Acting on a general superposition $|\Psi\rangle$ of computational basis states, $\mathbf{V}$ changes the sign of the component of the state along $|a\rangle$, while leaving unchanged the component orthogonal to $|a\rangle$:

$$\mathbf{V}|\Psi\rangle = |\Psi\rangle - 2|a\rangle\langle a|\Psi\rangle. \tag{4.7}$$

As we shall see, $\mathbf{U}_f$ is the only unitary transformation appearing in Grover's algorithm (where it appears repeatedly) that acts as anything other than the identity on the output register. Because the output register starts in the state $\mathbf{H}|1\rangle$, unentangled with the input register, and because $\mathbf{U}_f$ maintains the output register in this state, the output register remains unentangled with the input register and in the state $\mathbf{H}|1\rangle$ throughout Grover's algorithm. We could continue to describe things in terms of $\mathbf{U}_f$ and retain the 1-Qbit output register, expanding (4.7), for example, to the form

$$\mathbf{U}_f\Big(|\Psi\rangle \otimes \mathbf{H}|1\rangle\Big) = \big[|\Psi\rangle - 2|a\rangle\langle a|\Psi\rangle\big] \otimes \mathbf{H}|1\rangle. \tag{4.8}$$

But it is simpler to suppress all explicit reference to the unaltered output register — always unentangled with the input register and always in the state $\mathbf{H}|1\rangle$ — by replacing the $(n+1)$-Qbit unitary $\mathbf{U}_f$ by the n-Qbit unitary $\mathbf{V}$ that acts just on the n-Qbit input register, and defining all other operators that appear in the algorithm only by their action on the input register, with the implicit understanding that they act as the identity on the output register.

[1] This notation for projection operators is described in Section F of Chapter 1.

To execute Grover's algorithm, we once again initially transform the n-Qbit input register into the uniform superposition of all possible inputs:

$$|\phi\rangle = \mathbf{H}^{\otimes n}|0\rangle_n = \frac{1}{2^{n/2}} \sum_{x=0}^{2^n-1} |x\rangle_n. \tag{4.9}$$

It is now useful to give this state a name of its own: $|\phi\rangle$. In addition to $\mathbf{V}$, Grover's algorithm requires a second n-Qbit unitary $\mathbf{W}$ that acts on the input register in a manner similar to $\mathbf{V}$, but with a fixed form that does not depend on a. The unitary transformation $\mathbf{W}$ preserves the component of any state along the standard state $|\phi\rangle$, while changing the sign of its component orthogonal to $|\phi\rangle$:

$$\mathbf{W} = 2|\phi\rangle\langle\phi| - \mathbf{1}, \tag{4.10}$$

where $|\phi\rangle\langle\phi|$ is the projection operator on the state $|\phi\rangle$. We defer to section B below the not entirely obvious matter of how to build $\mathbf{W}$ out of 1-Qbit and 2-Qbit unitary gates.

Given implementations of $\mathbf{V}$ and $\mathbf{W}$, Grover's algorithm is quite straightforward. It consists of simply applying many times the product $\mathbf{WV}$ to the input register, taken initially to be in the state $|\phi\rangle$. If you prefer you can think of applying the $(n+1)$-Qbit transformations $\mathbf{U}_f = \mathbf{V} \otimes \mathbf{1}$ and $\mathbf{W} \otimes \mathbf{1}$ to the state $|\phi\rangle \otimes \mathbf{H}|1\rangle = \mathbf{H}^{\otimes(n+1)}\big(|0\rangle_n \otimes |1\rangle\big)$ of the combined input and output registers. But since both transformations leave unchanged the 1-Qbit output register in the state $|1\rangle$, disentangled from the input register, it is simpler to restrict attention to the action of $\mathbf{V}$ and $\mathbf{W}$ on the input register alone.

To see what is accomplished by repeatedly applying $\mathbf{WV}$ to the initial state $|\phi\rangle$, note that both $\mathbf{V}$ and $\mathbf{W}$ acting on either $|\phi\rangle$ or $|a\rangle$ give linear combinations of these two states. Since $\langle a|\phi\rangle = \langle\phi|a\rangle = 1/2^{n/2}$, independent of a, the linear combinations have real coefficients and are given by:

$$\mathbf{V}|a\rangle = -|a\rangle, \quad \mathbf{V}|\phi\rangle = |\phi\rangle - \tfrac{2}{2^{n/2}}|a\rangle;$$

$$\mathbf{W}|\phi\rangle = |\phi\rangle, \quad \mathbf{W}|a\rangle = \tfrac{2}{2^{n/2}}|\phi\rangle - |a\rangle. \tag{4.11}$$

So if we start with the state $|\phi\rangle$ and let any sequence of these two operators act successively, we will never be taken out of the two-dimensional plane spanned by real linear combinations of $|\phi\rangle$ and $|a\rangle$. Finding the result of repeated applications of $\mathbf{WV}$ to the initial state $|\phi\rangle$ reduces to an exercise in plane geometry.

It follows from the form (4.9) of $|\phi\rangle$, that $|\phi\rangle$ and $|a\rangle$, considered as vectors in the plane of their real linear combinations, are very nearly perpendicular, since the cosine of the angle γ between them is given by

$$\cos\gamma = \langle a|\phi\rangle = 2^{-n/2} \leq 1/\sqrt{N}, \tag{4.12}$$

which is small when N is large. It is convenient to define $|a_\perp\rangle$ to be the normalized real linear combination of $|\phi\rangle$ and $|a\rangle$ that is strictly orthogonal to $|a\rangle$ and makes the small angle $\theta = \pi/2 - \gamma$ with $|\phi\rangle$. (See Figures 4.2 and 4.3.) Since

$$\sin\theta = \cos\gamma = 2^{-n/2} \leq 1/\sqrt{N}, \tag{4.13}$$

θ is very accurately given by

$$\theta \approx 2^{-n/2} \tag{4.14}$$

when $\sqrt{N}$ is large.

Since **W** leaves $|\phi\rangle$ invariant and reverses the direction of any vector orthogonal to $|\phi\rangle$, its geometrical action on any vector in the two-dimensional plane containing $|\phi\rangle, |a\rangle$, and $|a_\perp\rangle$ is simply to replace the vector by its reflection in the mirror line through the origin along $|\phi\rangle$. On the other hand **V** reverses the direction of $|a\rangle$ while leaving any vector orthogonal to $|a\rangle$ invariant, so it acts on a general vector in the two-dimensional plane by replacing it with its reflection in the mirror line through the origin along $|a_\perp\rangle$. The product **WV**, being a product of two two-dimensional reflections, is a two-dimensional rotation.[2] The angle of that rotation is most easily seen by considering the effect of **WV** on $|a_\perp\rangle$ (see Figure 4.2). The application of **V** leaves $|a_\perp\rangle$ invariant, and the subsequent action of **W** on $|a_\perp\rangle$ reflects it in the line through the origin along the direction of $|\phi\rangle$. So the net effect of the rotation **WV** on $|a_\perp\rangle$ is to rotate $|a_\perp\rangle$ past $|\phi\rangle$ through a total angle that is twice the angle θ between $|a_\perp\rangle$ and $|\phi\rangle$.

Because **WV** is a rotation, the result of applying it to any other vector in the plane is also to rotate that vector through the angle 2θ in the direction from $|a_\perp\rangle$ to $|\phi\rangle$. So applying **WV** to the initial state $|\phi\rangle$ gives a vector rotated away from $|a_\perp\rangle$ by 3θ, since $|\phi\rangle$ is already rotated away from $|a_\perp\rangle$ by θ (Figure 4.3). Applying **WV** a second time results in a vector rotated away from $|a_\perp\rangle$ by 5θ, and each subsequent application of **WV** increases the angle between the final state and $|a_\perp\rangle$ by another 2θ. Since θ is very close to $2^{-n/2}$, after an integral number of applications as close as possible to

$$(\pi/4)2^{n/2}, \tag{4.15}$$

the resulting state will be very nearly orthogonal to $|a_\perp\rangle$ in the plane spanned by $|\phi\rangle$ and $|a\rangle$ — i.e. it will be very nearly equal to $|a\rangle$ itself.

Consequently a measurement of the input register in the computational basis will yield a with a probability very close to 1. We can check to see whether we have been successful

[2] A two-dimensional reflection can be achieved by adding a third dimension perpendicular to the plane and making 180° rotation with the mirror line as axis. (This reverses the irrelevant direction orthogonal to the plane.) The product of two such three-dimensional rotations is also a rotation, takes the plane into itself, and does not reverse the third orthogonal direction, so it is a two-dimensional rotation in the plane.

by "querying the oracle". If $f(a)$ is 1, as it will be with very high probability, this confirms that we have found the desired a. If we are very unlucky we might have to repeat the procedure a few more times before achieving success.

B. How to construct W

It remains to specify how to construct $\mathbf{W}$ out of 1-Qbit and 2-Qbit unitary gates. Now $-\mathbf{W}$ works just as well as $\mathbf{W}$ for purposes of the search algorithm, since it leads to a final state that differs, if at all, only by a harmless overall minus sign. It follows from (4.9) and (4.10) and the fact that $\mathbf{H}^{\otimes n}$ is its own inverse, that

$$-\mathbf{W} = \mathbf{1} - 2|\phi\rangle\langle\phi| = \mathbf{H}^{\otimes n}\big(\mathbf{1} - 2|00\ldots00\rangle\langle 00\ldots00|\big)\mathbf{H}^{\otimes n}, \tag{4.16}$$

so we need a gate that acts as the identity on every computational basis state except $|00\ldots00\rangle$, which it multiplies by -1. This is just the action of an $(n-1)$-fold controlled-$\mathbf{Z}$ gate, with the roles of the 1-Qbit states $|0\rangle$ and $|1\rangle$ interchanged. The interchange is accomplished by sandwiching the $(n-1)$-fold controlled-$\mathbf{Z}$ between $\mathbf{X}^{\otimes n}$ gates, and we therefore have

$$-\mathbf{W} = \mathbf{H}^{\otimes n}\mathbf{X}^{\otimes n}\big(\mathbf{c}^{n-1}\mathbf{Z}\big)\mathbf{X}^{\otimes n}\mathbf{H}^{\otimes n}. \tag{4.17}$$

We can construct $\mathbf{W}$ by constructing $\mathbf{c}^{n-1}\mathbf{Z}$, the $(n-1)$-fold controlled-$\mathbf{Z}$.

Figure 4.4 shows a straightforward but not terribly efficient way to make a $\mathbf{c}^{n-1}\mathbf{Z}$ gate for the case $n = 6$. We use $n-3$ ancillary Qbits, all initially in the state $|0\rangle$, $2(n-3)$ $\mathbf{c}^2\mathbf{X}$ (Toffoli) gates, and one $\mathbf{c}^2\mathbf{Z}$ gate. As noted in Chapter 2 (See Figures 2.11 and 2.12), these can all be built out of 1-Qbit and 2-Qbit gates. It is essential for the success of the algorithm that each ancillary Qbit be restored to its initial state $|0\rangle$, since our analysis of the Grover algorithm assumes that the input and output registers have states of their own, unentangled with any other Qbits after each application of $\mathbf{W}$ and $\mathbf{V}$.

The construction of Figure 4.4 is rather expensive in Qbits, requiring $n-3$ ancillas to apply the algorithm to an n-bit set of possibilities for the special number a. At a cost of four times as many Toffoli gates, one can reduce the number of ancillas to a single one, regardless of the size of n. The way to do this is developed in Figures 4.5-4.7. Figures 4.5 and 4.6 show how nearly doubling the number of gates makes it possible for the construction of Figure 4.4 to work for *arbitrary* initial states of the ancillas. Figure 4.7 then shows how by an additional doubling one can, with the aid of a single ancilla, divide up an n-fold controlled-$\mathbf{Z}$ into two multiply controlled NOT gates and two multiply controlled-$\mathbf{Z}$ gates, each acting on about $\frac{1}{2}n$ Qbits. (Since $\mathbf{X} = \mathbf{HZH}$, one can convert a multiply controlled-$\mathbf{Z}$ gate to a multiply controlled-NOT gate by applying Hadamard gates to the target Qbit at the beginning and end of the circuit.) The controlled-$\mathbf{Z}$ gates are able nondisruptively to use the control Qbits of the controlled-NOT gates as their ancillary Qbits in the construction of Figure 4.5, and the controled-NOT gates can make similar use of the control Qbits of the controlled-$\mathbf{Z}$ gates.

C. Generalization to several special numbers.

If there are several special numbers, essentially the same algorithm can be used to find one of them, provided we know how many there are. The function f in (4.1) now becomes

$$f(x) = 0,\ x \neq a_1, \ldots a_m; \qquad f(x) = 1,\ x = a_1, \ldots a_m. \tag{4.18}$$

The n-Qbit unitary transformation $\mathbf{V}$ extracted from (4.4) becomes one whose action on computational basis states in the input register is given by

$$\mathbf{V}|x\rangle = |x\rangle,\ x \neq a_1, \ldots a_m; \qquad \mathbf{V}|x\rangle = -|x\rangle,\ x = a_1, \ldots a_m. \tag{4.19}$$

If we replace the state $|a\rangle$ by

$$|\psi\rangle = \frac{1}{\sqrt{m}} \sum_{i=1}^{m} |a_i\rangle,$$

then starting with $|\phi\rangle$, which continues to have the form (4.9), the transformations $\mathbf{V}$ and $\mathbf{W}$ now keep the state of the input register in the two-dimensional plane spanned by the real linear combinations of $|\psi\rangle$ and $|\phi\rangle$. The unitary transformation $\mathbf{V}$ changes the sign of $|\psi\rangle$ but preserves the linear combination of $|\phi\rangle$ and $|\psi\rangle$ orthogonal to $|\psi\rangle$, so $\mathbf{V}$ is now a reflection in the line through the origin along the vector $|\psi_\perp\rangle$ perpendicular to $|\psi\rangle$ in the plane. Everything else is just as in the case of a single special number except that now the angle Θ between $|\psi_\perp\rangle$ and $|\phi\rangle$ satisfies

$$\sin\Theta = \cos(\pi/2 - \Theta) = \langle\psi|\phi\rangle = \sqrt{m/2^n}. \tag{4.20}$$

When $m/2^n \ll 1$, we can arrive at a state very close to $|\psi\rangle$ with

$$(\pi/4)2^{n/2}/\sqrt{m} \tag{4.21}$$

applications of $\mathbf{WV}$. A measurement then gives us, with a probability very close to 1, a random one of the special values a_i. Note that the mean number of invocations of the subroutine only decreases as $1/\sqrt{m}$ with the number m of marked items, in contrast to a classical search, where doubling the number of acceptable solutions would halve the time of the search. When $m/2^n$ is not small we have to reexamine the expression (4.21) for the optimal number of iterations, but at that point the quantum search offers little significant advantage over a classical one.

We must know how many special numbers there are for the procedure to work, since we have to know how many times to do the Grover iteration before making our measurement. By exploiting the fact that the Grover iteration is periodic, restoring the initial state after about $\pi 2^n/\sqrt{m}$ iterations, it is possible to combine Grover iterations with a clever application of the quantum Fourier transform to learn the value of m with enough accuracy to enable one then to apply the Grover iteration the right number of times to ensure a high probability of success.

Figure 4.1

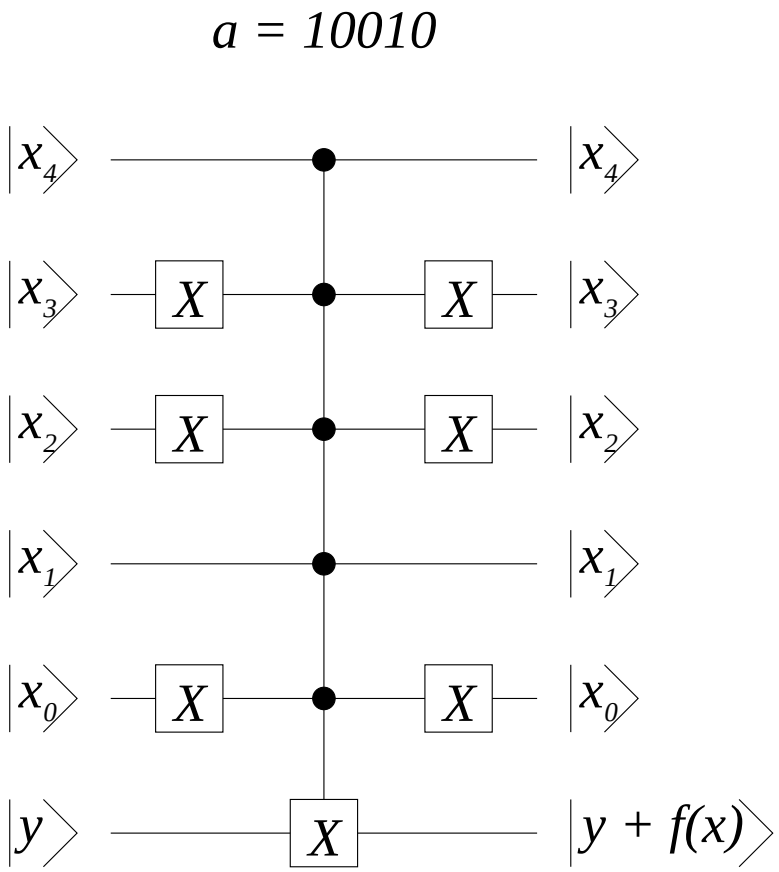

Figure 4.1. A possible realization of a black box that executes the unitary transformation $\mathbf{U}_f\Big(|x\rangle_n|y\rangle_1\Big) = |x\rangle_n|y \oplus f(x)\rangle_1$, where $f(x) = 0,\ x \neq a;\ f(x) = 1,\ x = a$. The input register has $n = 5$ Qbits and the special number a is 10010. The 6-Qbit gate in the center of the figure is a five-fold controlled-NOT, which acts on the computational basis to flip the target bit if and only if every one of the five control bits is in the state $|1\rangle$. Ways to construct such a gate out of more elementary gates are shown in Figures 4.4-4.7.

Figure 4.2

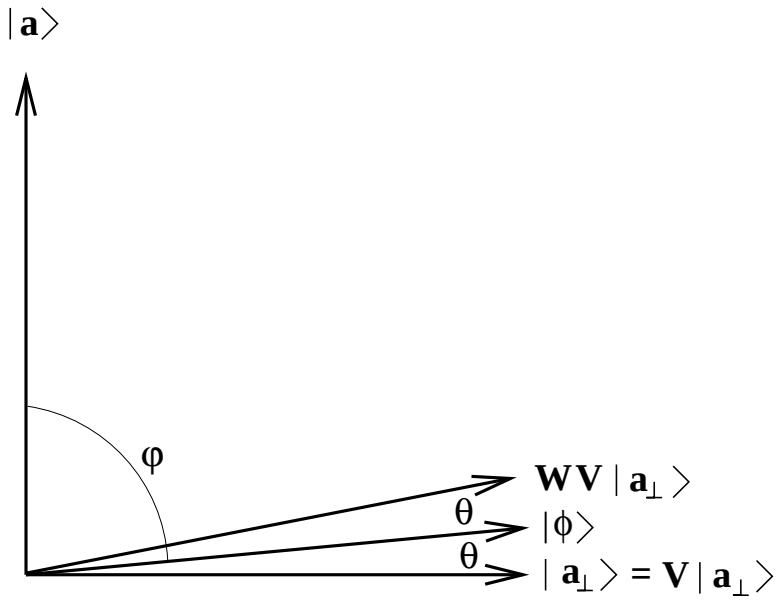

Figure 4.2. Real linear combinations of the special state $|a\rangle$, and the uniform superposition $|\phi\rangle = 2^{-n/2}\sum|x\rangle$, define a plane in which these two states are very nearly orthogonal. The state $|a_\perp\rangle$ in that plane, making a small angle θ with $|\phi\rangle$, is exactly orthogonal to $|a\rangle$. The unitary transformation **V** takes any vector in the plane into its reflection in the line through the origin along $|a_\perp\rangle$, so it leaves $|a_\perp\rangle$ invariant. The unitary transformation **W** takes any vector in the plane into its reflection in the line through the origin along $|\phi\rangle$, so it rotates $|a_\perp\rangle$ counterclockwise through the angle 2θ. Therefore the combined rotation **WV** rotates $|a_\perp\rangle$ counterclockwise through 2θ, and since **WV** is indeed a rotation it does the same to any vector in the plane.

Figure 4.3

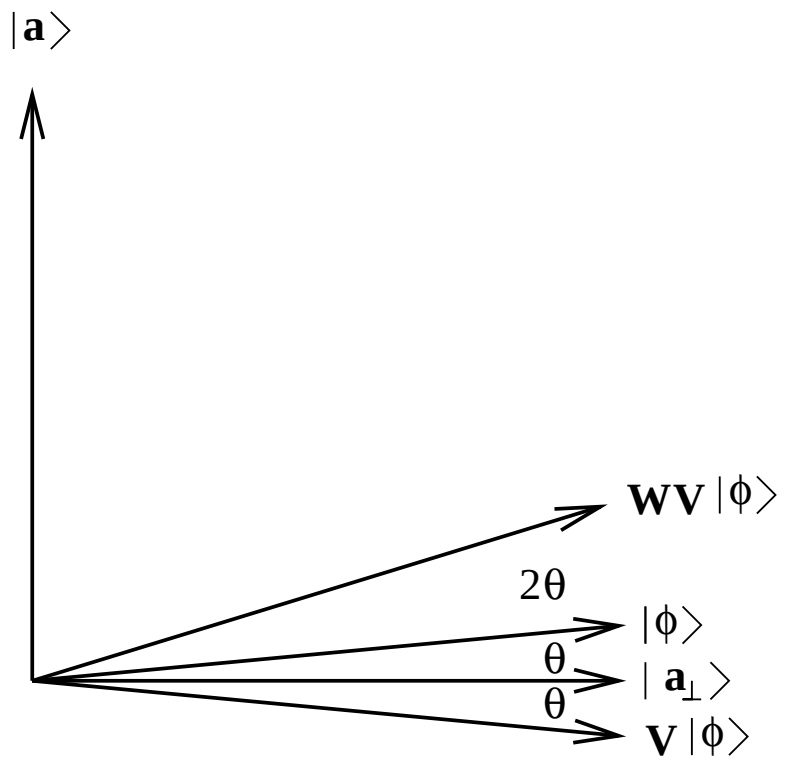

Figure 4.3. Since the rotation **WV** rotates any vector in the plane of real linear combinations of $|a\rangle$ and $|\phi\rangle$ clockwise through an angle 2θ, it takes $|\phi\rangle$ into a vector $\mathbf{WV}|\phi\rangle$ that makes an angle 3θ with $|a_\perp\rangle$. This can also be seen directly fr om the separate behaviors of **V** and **W**: **V** takes $|\phi\rangle$ into its mirror image in $|a_\perp\rangle$, and **W** then takes $\mathbf{V}|\phi\rangle$ into its mirror image in $|\phi\rangle$.

Figure 4.4

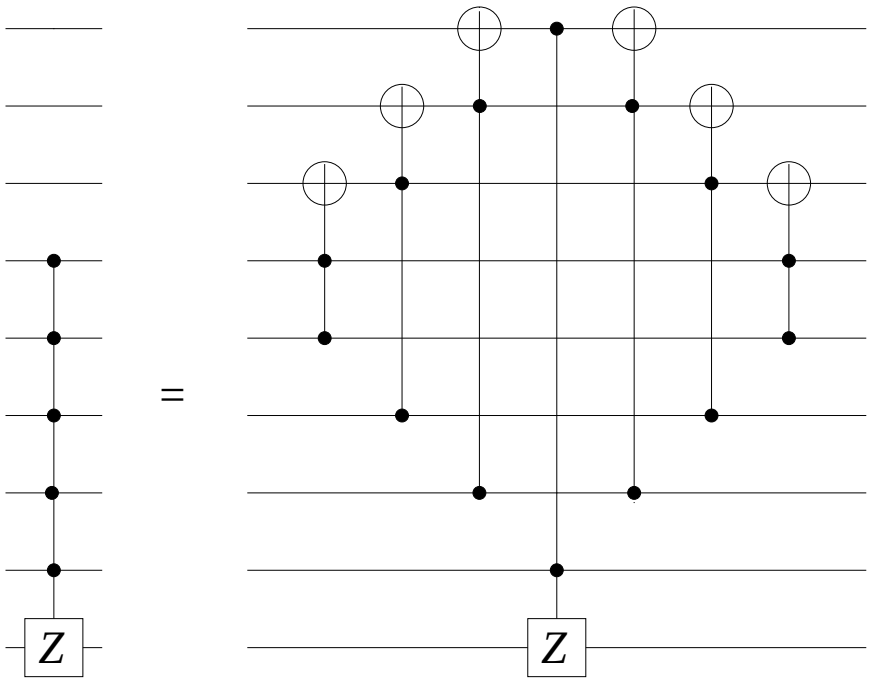

Figure 4.4. The n-fold controlled-$\mathbf{Z}$ transformation, $\mathbf{c}^n\mathbf{Z}$, acts as the identity on states of the computational basis unless all n control Qbits are in the state $|1\rangle$, when it acts on the target Qbit as $\mathbf{Z}$. It can be constructed if one has an additional $n-2$ ancilliary Qbits, all initially in the state $|0\rangle$. One uses $2(n-2)$ $\mathbf{c}^2\mathbf{X}$ (Toffoli) gates and a $\mathbf{c}^2\mathbf{Z}$ gate. The construction is illustrated for the case $n = 5$. The top 3 wires are the 3 ancilliary Qbits. The next 5 wires from the top are the 5 control Qbits, and the bottom wire is the target Qbit. One easily verifies (applying the circuit to computational basis states, with each of the ancilliary Qbits in the state $|0\rangle$) that $\mathbf{Z}$ acts on the target Qbit if and only if every one of 5 control Qbits is in the state $|1\rangle$. The Toffoli gates are symmetrically disposed on both sides of the diagram to ensure that at the end of the process each of the 3 ancilliary Qbits is set back to its initial state $|0\rangle$. This is essential if the ancilliary Qbits are not to become entangled with the Qbits on which the Grover iteration acts, represented by the bottom 6 wires.

Figure 4.5

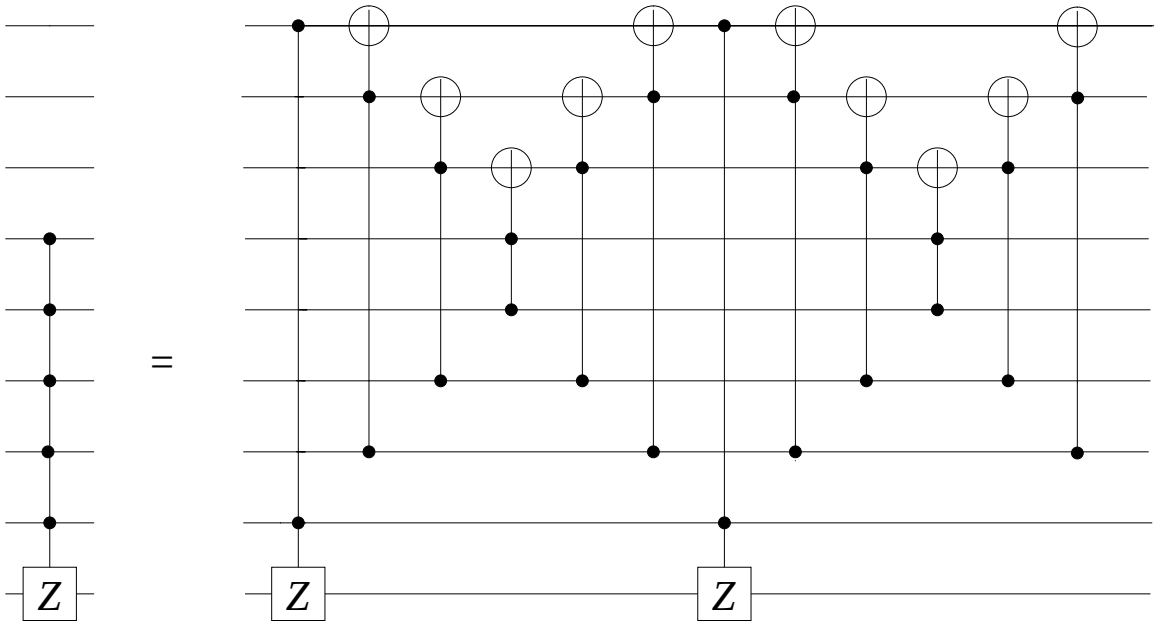

Figure 4.5. An improved version of Figure 4.4, with twice as many gates. Gates have been added on the left and right to ensure that the circuit works for *arbitrary* initial computational basis states of the 3 ancillary Qbits at the top, restoring them to their initial states at the end of the computation. To see this note, first that because Toffoli gates or $\mathbf{c}^2\mathbf{Z}$ gates are their own inverses, the circuit acts as the identity on those computational basis states of all 9 Qbits in which any one of the 5 control Qbits (2nd through 6th wires from the bottom) is in the state $|0\rangle$, regardless of the computational-basis states of the other Qbits. This is because, as an examination of the figure reveals, replacing the gate governed by any one of the 5 control Qbits with the identity always results in a pairwise cancellation of all the remaining gates. It remains only to confirm that when all 5 control Qbits are in the state $|1\rangle$, the circuit acts as $\mathbf{Z}$ on the target Qbit at the bottom, and the state of the three ancillary Qbits at the top is unchanged. This is established in Figure 4.6, which shows the operation of the gates in Figure 4.5 when the 5 control Qbits are all in the state $|1\rangle$. Note that because $\mathbf{X} = \mathbf{HZH}$ one can use this circuit to produce a multiply controlled-NOT gate by applying Hadamard gates to the bottom wire on the far right and left.

Figure 4.6

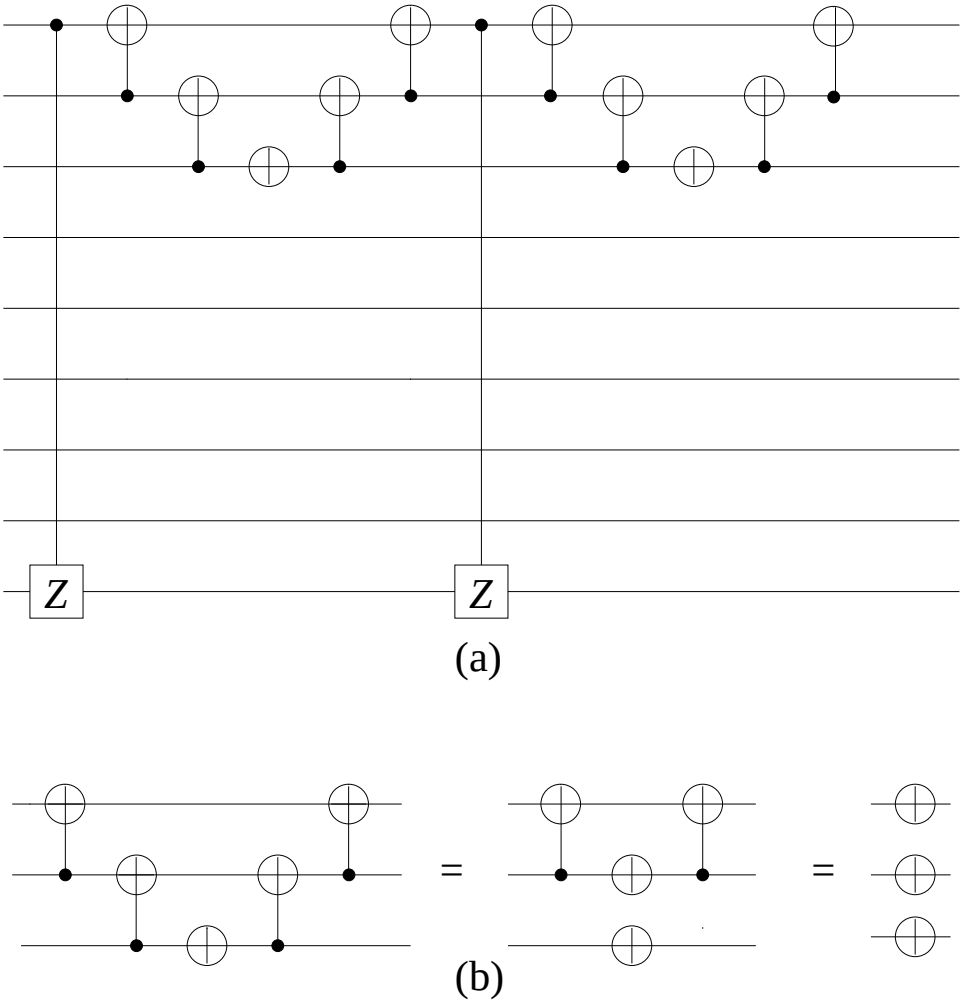

Figure 4.6. Part (a) reproduces what remains of Figure 4.5 when all five control Qbits are in the state $|1\rangle$. One easily verifies that two identical cNOT gates, separated by a NOT acting on their control Qbit, have exactly the same action on the computational basis as a pair of NOT gates that act on both the control and target Qbit. As a result each of the two identical sets of 5 adjacent gates acting on the three ancillary Qbits at the top of part (a) (Since $\mathbf{X} = \mathbf{HZH}$ Figure 4.5 works for either type.) reduces simply to three NOT gates, as shown in part (b). Making this further simplification in part (a), note that because each of the three ancillary Qbits is acted on by two NOT gates, its state is unaltered. The two NOT gates acting on the upper wire also ensure that precisely one of the two **cZ** gates applies **Z** to the bottom Qbit, independent of the state of the upper wire.

Figure 4.7

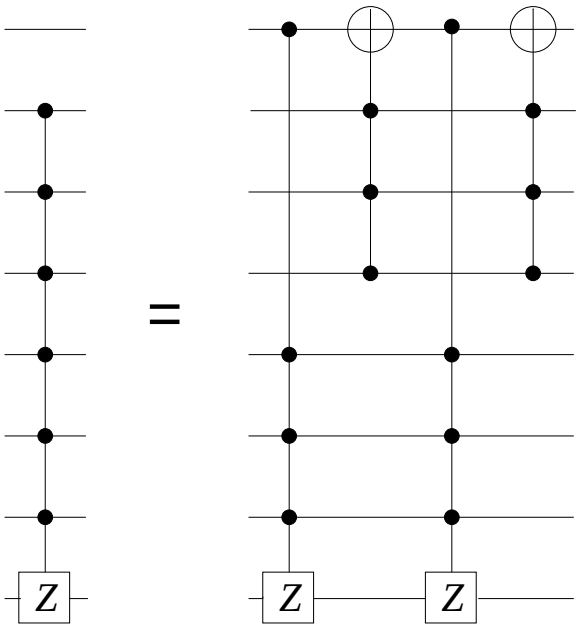

Figure 4.7. The identity illustrated by the circuit is easily confirmed. There is only one ancilla, whose state is left unchanged. By introducing circuits of the form in Figure 4.5 into this circuit one can produce $\mathbf{c}^n\mathbf{Z}$ or $\mathbf{c}^n\mathbf{X}$ gates with the aid of just a single ancilla. The point is that in constructing each of the multiply controlled-NOT gates in Figure 4.7 out of Toffoli gates, as prescribed in Figure 4.5, one can borrow the control Qbits of the multiply controlled-$\mathbf{Z}$ gates to use as ancillary Qbits in the expansions of Figure 4.5, since those expansions work regardless the state of their ancillary Qbits, and restore it to its original form. For the same reasons one can also borrow the control Qbits of the multiply controlled-NOT gates to construct the multiply controlled-$\mathbf{Z}$ gates.