Chapter 5

Quantum error correction

5.1 The miracle of quantum error correction

Correcting errors might sound like a dreary practical problem, of little aesthetic or conceptual interest. But aside from being of crucial importance for the feasibility of quantum computation, it is also one of the most beautiful and surprising parts of the subject. The surprise is that error correction is possible at all, since the only way to detect errors is to make measurements, but measurement gates disruptively alter the states of the measured Qbits, apparently making things even worse. "Quantum error correction" would seem to be an oxymoron. The beauty lies in the ingenious ways that people have found to get around this apparently insuperable obstacle.

The discovery in 1995 of quantum error correction by Peter Shor and, independently, Andrew Steane had an enormous impact on the prospects for actual quantum computation. It changed the dream of building a quantum computer capable of useful computation from a clearly unattainable vision to a program that poses an enormous but not necessarily insuperable technological challenge.

Error correction is not a major issue for classical computation. In a classical computer the physical systems that embody individual bits – the Cbits – are immense on the atomic scale. The two states of a Cbit representing 0 and 1 are so grossly different that the probability is infinitesimal for flipping from one to the other as a result of thermal fluctuations, mechanical vibrations, or other irrelevant extraneous interactions.

Error correction does become important, even classically, in the transmission of information over large distances, because the farther the signal travels, the more it attenuates. One can deal with this in a variety of straightforward or ingenious ways. One of the crudest is to encode each logical bit in three actual bits, replacing $|0\rangle$ and $|1\rangle$ by the codewords

$$|\overline{0}\rangle = |0\rangle|0\rangle|0\rangle = |000\rangle, \qquad |\overline{1}\rangle = |1\rangle|1\rangle|1\rangle = |111\rangle. \tag{5.1}$$

One can then monitor each codeword, checking for flips in any of the individual Cbits and restoring them by applying the principle of majority rule whenever a flip is detected. Monitoring has to take place often enough to make negligible the probability that more than a single bit has flipped in a single codeword between inspections.

Quantum error correction also uses multi-Qbit codewords and also requires monitoring at a rate that renders certain kinds of compound errors highly improbable. But there are several ways in which error correction in a quantum computer is quite different.

(a) A quantum computer, unlike a classical computer, requires error correction. The physical Qbits are individual atomic-scale physical systems such as atoms, photons, trapped ions, or nuclear magnetic moments. Any coupling to anything not under the explicit control of the computer and its program can substantially disrupt the state associated with those Qbits, entangling them with computationally irrelevant features of the computer or the world outside the computer, thereby destroying the computation. For a quantum computer to work without error correction, each Qbit would have to be impossibly well isolated from irrelevant interactions with other parts of the computer and anything else in its environment.

(b) In contrast to classical error correction, checking for errors in a quantum computer is problematic. The obvious way to monitor a Qbit is to measure it. But the result of measuring a Qbit is to alter its state, if it has one of its own, and, more generally, to destroy its quantum correlations with other Qbits with which it might be entangled. Such disruptions are stochastic – i.e. unpredictable – and introduce major errors of their own. One must turn to less obvious forms of monitoring.

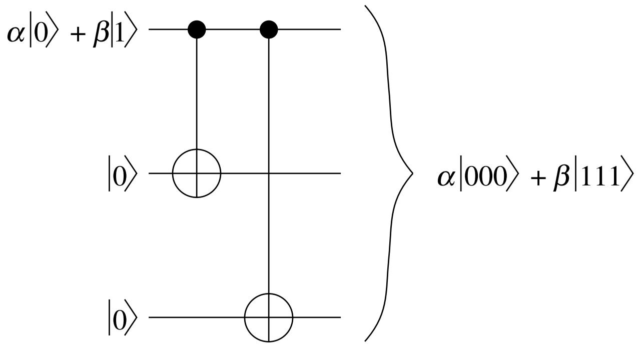
(c) Bit flips are not the only errors. There are entirely nonclassical sources of trouble. For example phase errors, such as the alteration of $|0\rangle + |1\rangle$ to $|0\rangle - |1\rangle$, can be just as damaging.

(d) Unlike the discrete all-or-nothing bit-flip errors suffered by Cbits, errors in the state of Qbits grow continuously out of their uncorrupted state.

We begin our discussion of error correction by examining in Section 5.2 a simple model of quantum error correction that works when the possible errors are artificially limited to a few specific kinds of disruption. Although this is clearly unrealistic, the error-correction procedure is easy to follow. It also introduces in a simple setting most of the tricks that continue to work in the more realistic case.

5.2 A simplified example

Much of the flavor of quantum error correction is conveyed by an artificially simple model in which the only errors a collection of Qbits is allowed to experience are the classically meaningful errors: random



$\alpha|0\rangle+\beta|1\rangle$
$|0\rangle$
$|0\rangle$
$\alpha|000\rangle+\beta|111\rangle$

Fig 5.1 A unitary circuit that encodes the 1-Qbit state $\alpha|0\rangle + \beta|1\rangle$ into the 3-Qbit code state $\alpha|000\rangle + \beta|111\rangle$, using two cNOT gates and two other Qbits each initially in the state $|0\rangle$. The circuit clearly works for the computational-basis states $|0\rangle$ and $|1\rangle$, and therefore, by linearity, it works for arbitrary superpositions.

flips of individual Qbits. We shall examine the more general possibilities for quantum errors in Section 5.3 below.

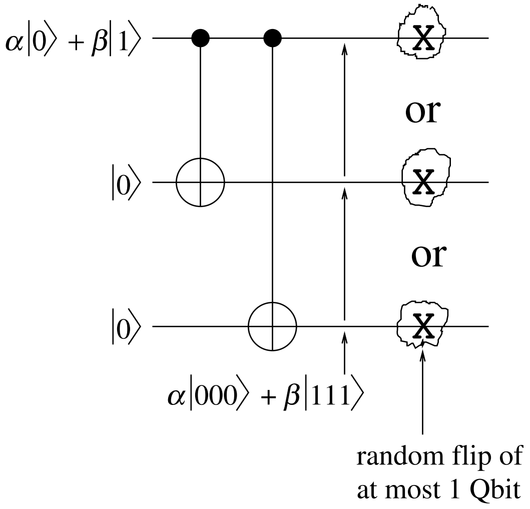
Bit-flip errors in a computation can be modeled by a circuit that differs from the ideal error-free circuit only in the occasional presence of extraneous 1-Qbit NOT gates. If such randomly occurring error-producing NOT gates are sufficiently rare, then since the only allowed errors are bit-flip errors, one might hope to be able to correct the corruptions they introduce by tripling the number of Qbits and using precisely the 3-Qbit code (5.1) that corrects for bit-flip errors in the classical case. Because of the disruptive effect of measurement gates in diagnosing errors, it is not at all clear that such a 3-Qbit code can be effective for bit-flip errors in the quantum case. It can nevertheless be made to work, though the way in which one does the encoding and performs the error correction is much subtler for Qbits than it is for Cbits.

To begin with, there is the question of encoding. Classically one merely replaces each of the two computational-basis states $|x\rangle$ by the codeword states $|\overline{x}\rangle = |x\rangle|x\rangle|x\rangle$, for $x = 0$ or 1. Qbits, however, can also be in superpositions $\alpha|0\rangle + \beta|1\rangle$, and one requires a circuit that automatically encodes this into $\alpha|\overline{0}\rangle + \beta|\overline{1}\rangle = \alpha|0\rangle|0\rangle|0\rangle + \beta|1\rangle|1\rangle|1\rangle$ for arbitrary α and β, in the absence of any knowledge of what the values of α and β might be. This can be done with two cNOT gates that target two additional Qbits initially both in the state $|0\rangle$, as illustrated in Figure 5.1:

$$\alpha|\overline{0}\rangle + \beta|\overline{1}\rangle = \alpha|0\rangle|0\rangle|0\rangle + \beta|1\rangle|1\rangle|1\rangle = \mathbf{C}_{21}\mathbf{C}_{20}\big(\alpha|0\rangle + \beta|1\rangle\big)|0\rangle|0\rangle. \tag{5.2}$$

Having produced such a 3-Qbit codeword state, we must then guard against its corruption by the possible action of an extraneous NOT gate that acts on at most one of the three Qbits, as illustrated in Figure 5.2. This is easily done for Cbits, for which there are only two possible uncorrupted initial states, $|000\rangle$ and $|111\rangle$, and examining them is unproblematic. To see whether either initial state has been corrupted by the action of a single NOT gate, one nondisruptively reads

Fig 5.2 The encoded state of Figure 5.1 may or may not be corrupted by the action of a single extraneous NOT gate. The error-inducing gates are depicted in a lighter font – X instead of **X** – and inside a noisy-looking corrupted box.


$\alpha|0\rangle + \beta|1\rangle$
$|0\rangle$
$|0\rangle$
X
or
X
or
X
$\alpha|000\rangle + \beta|111\rangle$
random flip of
at most 1 Qbit

the three Cbits. If this reveals all three Cbits to be in the same state, there is no corruption to correct. If one of them is found to be in a different state from the other two, that particular Cbit is the one that was acted upon by the extraneous NOT gate, and applying a second NOT gate to that Cbit restores the initial state.

One cannot, however, nondisruptively "read" the state of a collection of Qbits. The only way to extract information is by the action of measurement gates. But measuring any of the three Qbits immediately destroys the uncorrupted superposition

$$|\Psi\rangle = \alpha|000\rangle + \beta|111\rangle, \tag{5.3}$$

converting it either to $|000\rangle$ (with probability $|\alpha|^2$) or to $|111\rangle$ (with probability $|\beta|^2$). There is a similar coherence-destroying effect on each of the three possible corrupted states,

$$\begin{aligned} |\Psi_0\rangle &= \mathbf{X}_0|\Psi\rangle = \alpha|001\rangle + \beta|110\rangle, \\ |\Psi_1\rangle &= \mathbf{X}_1|\Psi\rangle = \alpha|010\rangle + \beta|101\rangle, \\ |\Psi_2\rangle &= \mathbf{X}_2|\Psi\rangle = \alpha|100\rangle + \beta|011\rangle, \end{aligned} \tag{5.4}$$

obliterating any dependence of the post-measurement state on the complex amplitudes α and β. This might appear (and for some time was thought) to be the end of the story: quantum error correction is impossible because of the disruptive effect of the measurement needed to diagnose the error.

But there are subtler ways to extract the information needed to diagnose and correct possible errors. Although there continues to be a disruption in these refined procedures, the damaging effects are shifted from the codeword Qbits to certain ancillary Qbits. By coupling the codeword Qbits to these ancillary Qbits with appropriate 2-Qbit unitary gates, and then applying measurement gates only to the ancillas, one can extract information about certain *relations* prevailing among the codeword Qbits. This more limited information turns out to be enough to diagnose and correct certain errors in a coherence-preserving manner, without revealing anything about the original uncorrupted codeword state. Acquiring no information about the uncorrupted state is a necessary restriction on any error-correction procedure capable of perfectly restoring the uncorrupted state. If one could get even partial information about the structure of a state without disrupting it, one could continue collecting additional information nondisruptively until one was well on the way to violating the no-cloning theorem.

Note that all possible forms for the uncorrupted 3-Qbit codeword (5.3) – given by assigning all possible values to the amplitudes α and β – lie in a two-dimensional subspace of the full eight-dimensional space containing all possible 3-Qbit states. Furthermore, each of the three allowed corruptions (5.4) also lies in a two-dimensional subspace of the full 3-Qbit space, and the three subspaces containing the three allowed corruptions are each orthogonal to the subspace containing the uncorrupted codeword, and orthogonal to each other. This turns out to be crucial to the success of the enterprise.

More generally, if we wanted to use an n-Qbit codeword in a model in which the only allowed errors were flips of a single Qbit, then we would require $2(1 + n)$ dimensions to accommodate the $n + 1$ mutually orthogonal two-dimensional subspaces associated with a general uncorrupted state and its n different 1-Qbit corruptions. Since all possible states of n Qbits span a 2^n-dimensional space, a necessary condition for an n-Qbit bit-flip-error-correcting code to be possible is

$$2^{n-1} \geq 1 + n. \tag{5.5}$$

The smallest n satisfying (5.5) is $n = 3$, for which it holds as an equality. This shows that the 3-Qbit code is, in this sense, perfect for the purpose of correcting errors limited to flips of a single Qbit.

Figure 5.3 shows that 3-Qbit codewords, as well as meeting this necessary condition for the correction of quantum bit-flip errors, actually do permit it to be carried out. The error detection and correction requires two additional ancillary Qbits (the upper two wires), initially both in the state $|0\rangle$. Both ancillas are targeted by pairs of cNOT gates controlled by subsets of the three codeword Qbits. Note first that if the 3-Qbit codeword has not been corrupted, so its state remains (5.3), then both the ancillary Qbits remain in the state $|0\rangle$ after the action of the cNOT gates, since the term $|000\rangle$ in the codeword results in none of the target Qbits being flipped, while the term $|111\rangle$ results in both of the target Qbits being flipped twice, which is equivalent to no flip.

In a similar way each of the three corruptions (5.4) results in a different unique final state for the ancillary Qbits. The first of those

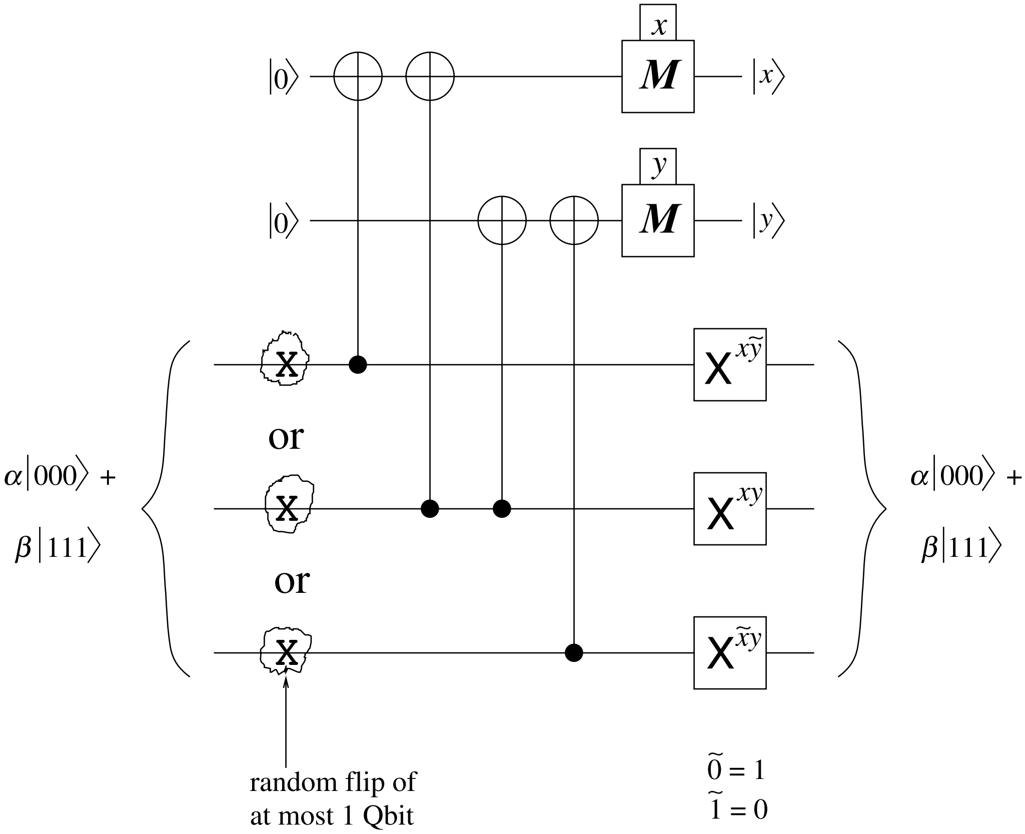


Fig 5.3 How to detect and correct the three possible single-bit-flip errors shown in Figure 5.2. One requires two ancillary Qbits (the upper two wires), each initially in the state $|0\rangle$, coupled to the codeword Qbits by cNOT gates. After the cNOT gates have acted each ancilla is measured. If both measurements give 0, then none of the erroneous NOT gates on the left have acted and none of the error-correcting NOT gates on the right need to be applied. If the upper measurement gate shows $x = 1$ and the lower one shows $y = 0$, then the uppermost of the three erroneous NOT gates has acted on the left. Its action is undone by applying the uppermost of the three NOT gates on the right. The other two possible 1-Qbit errors are similarly corrected.

corruptions results in $|0\rangle$ for the upper ancilla and $|1\rangle$ for the lower, since either term in the superposition $\alpha|001\rangle + \beta|110\rangle$ results in zero or two flips for the upper ancilla, and a single flip for the lower ancilla. The next form in (5.4) produces a single flip for both ancillas, resulting in $|1\rangle$ for both. The third results in $|1\rangle$ for the upper and $|0\rangle$ for the lower ancilla.

So if the two ancillary Qbits are measured after the cNOT gates have acted, the four possible readings, 00, 01, 10, and 11, of the two measurement gates reveal whether or not a random one of the codeword Qbits has been flipped and, in the event of a flip, which of the three has suffered it. On the basis of this information one can either accept the codeword as uncorrupted or apply a NOT gate to the Qbit that has been identified as the flipped one, thereby restoring the initial uncorrupted state. One easily confirms that this is precisely what is accomplished by the NOT gates on the extreme right of Figure 5.3.

This accomplishes what any valid quantum error-correction procedure must do: it restores the original uncorrupted state without revealing any information whatever about what the form of that state – the particular values of the amplitudes α and β – might actually be. The procedure succeeds in preserving the superposition by extracting information only about correlations among the Qbits making up the codeword, without ever extracting information about individual Qbits. Working only with correlations makes it possible to apply a single linear test that works equally well for diagnosing 1-Qbit errors in either $|000\rangle$ or $|111\rangle$, and therefore also works for any superposition of those states.

This simple example of quantum error correction requires the use of measurement gates to diagnose the error. The outputs of the measurement gates are noted, and then used to determine which, if any, of a collection of error-correcting NOT gates should be applied. The procedure can be automated into a bigger quantum circuit that eliminates (or almost eliminates) the need to use measurement gates combined with unitary gates, which are or are not applied depending on the readings of the measurement gates. This can be achieved by a combination of cNOT and Toffoli gates, controlled by the ancillary Qbits, as shown in Figure 5.4.

Replacing measurement gates by additional cNOT gates does not entirely eliminate the need for measurement, because at the end of the process the state of the ancillary Qbits will depend on the character of the error and will in general no longer be the state $|0\rangle|0\rangle$ with which the error-correction procedure starts. If one wishes to reuse these ancillary Qbits for further error correction, it is necessary to reset them to their initial state $|0\rangle|0\rangle$. This can efficiently be done by *measuring* them and applying the appropriate NOT gates if either is found to be in the state $|1\rangle$. Thus measurement gates followed by NOT gates that depend on the measurement outcome are still needed to prepare the circuit for a possible future error correction.

This procedure (automated or not) will continue to work even when the codeword Qbits are entangled with many other codeword Qbits, as they will be in the course of a nontrivial computation. In such a case the codeword Qbits have no state of their own, the state of all the many codeword Qbits being of the form

$$\alpha|000\rangle|\Psi\rangle + \beta|111\rangle|\Phi\rangle, \tag{5.6}$$

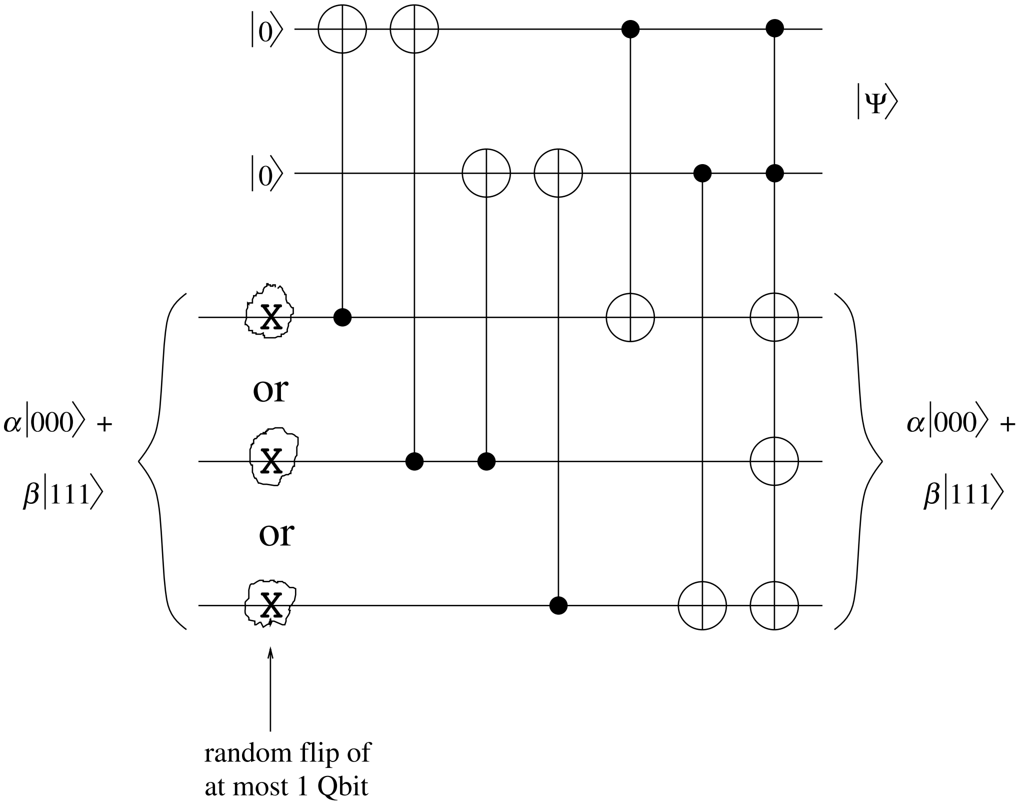


Fig 5.4 Automation of the error-correction process of Figure 5.3. The three controlled gates on the right – one of them a doubly controlled Toffoli gate with multiple targets – have precisely the same error-correcting effect on the three codeword Qbits as does the application of NOT gates contingent on measurement outcomes in Figure 5.3. The final state $|\Psi\rangle$ of the ancillas (which is also the state that determines the action of the three controlled gates on the right) is $|00\rangle$ if none of the erroneous NOT gates on the left has acted. It is $|10\rangle$ if only the upper erroneous NOT gate has acted, $|11\rangle$ if only the middle one has acted, and $|01\rangle$ if only the lower one has acted.

with the error correction applied to the three Qbits on the left. One easily confirms that the added complication of entanglement with other Qbits has no effect on the validity of the error-correction procedure.

There is an alternative way of representing the use of cNOT gates in Figure 5.3 to diagnose the error, which is useful in correcting quantum errors in more realistic cases. The alternative point of view is based on the easily confirmed fact that the uncorrupted state (5.3) is left unaltered by either of the operators $\mathbf{Z}_2\mathbf{Z}_1$ and $\mathbf{Z}_1\mathbf{Z}_0$, while the three corruptions (5.4) are each eigenstates of $\mathbf{Z}_2\mathbf{Z}_1$ and $\mathbf{Z}_1\mathbf{Z}_0$ with distinct

Table 5.1. Two operators that diagnose the error syndrome for the 3-Qbit code that protects against bit-flip errors. The four entries in each of the two rows indicate whether the operator for that row commutes (+) or anticommutes (−) with the operators at the top of the four columns

	$\mathbf{X}_2$	$\mathbf{X}_1$	$\mathbf{X}_0$	$\mathbf{1}$
$\mathbf{Z}_2\mathbf{Z}_1$	−	−	+	+
$\mathbf{Z}_1\mathbf{Z}_0$	+	−	−	+

sets of eigenvalues: 1 and −1 in the case of $|\Psi_0\rangle$; −1 and −1 in the case of $|\Psi_1\rangle$; and −1 and 1 in the case of $|\Psi_2\rangle$.

While these last three facts can be confirmed directly from the explicit forms of $|\Psi_0\rangle$, $|\Psi_1\rangle$, and $|\Psi_2\rangle$ on the right of (5.4), it is worth noting, for purposes of comparison with some of the more complex cases that follow, that they also follow from the facts that $\mathbf{Z}_2\mathbf{Z}_1$ and $\mathbf{Z}_1\mathbf{Z}_0$ act as the identity on the uncorrupted state $|\Psi\rangle$, that the corrupted states are of the form $|\Psi_j\rangle = \mathbf{X}_j|\Psi\rangle$, and that $\mathbf{X}_j$ *commutes* with $\mathbf{Z}_i$ when $i \neq j$, while $\mathbf{X}_j$ *anticommutes* with $\mathbf{Z}_j$: $\mathbf{Z}_j\mathbf{X}_j = -\mathbf{X}_j\mathbf{Z}_j$. The resulting pattern of commutations (+) or anticommutations (−) is summarized in Table 5.1.

Thus the joint eigenvalues of the commuting operators $\mathbf{Z}_2\mathbf{Z}_1$ and $\mathbf{Z}_1\mathbf{Z}_0$ distinguish among the uncorrupted state and each of the three possible corruptions. A procedure that takes advantage of this by sandwiching controlled $\mathbf{Z}_2\mathbf{Z}_1$ and controlled $\mathbf{Z}_1\mathbf{Z}_0$ gates between Hadamards acting on the control Qbits is shown in Figure 5.5. Although it takes a little thought to confirm directly from the figure that Figure 5.5 does indeed accomplish error correction – we shall work this out in Section 5.4 as a special case of a much more general procedure – one can confirm that it does by simply noting that Figure 5.5 is mathematically equivalent to Figure 5.3. This equivalence follows from the facts that $\mathbf{X} = \mathbf{HZH}$, that $\mathbf{H}^2 = \mathbf{1}$, and that the action of controlled-Z is unaltered by exchanging the target and control Qbits.

This oversimplified example, in which only bit-flip errors are allowed, illustrates most of the features of quantum error correction that one encounters in more realistic cases. The more general procedure is complicated by the fact that, as noted above and made precise in Section 5.3 below, the general error a Qbit can experience is more complicated than just a single bit flip. As a result, one needs codewords containing more than three Qbits to correct general single-Qbit errors, and one requires more complicated diagnostic and corrective procedures than those of Figures 5.3–5.5, involving more than just a pair of ancillary Qbits. But although the codewords and error-correcting

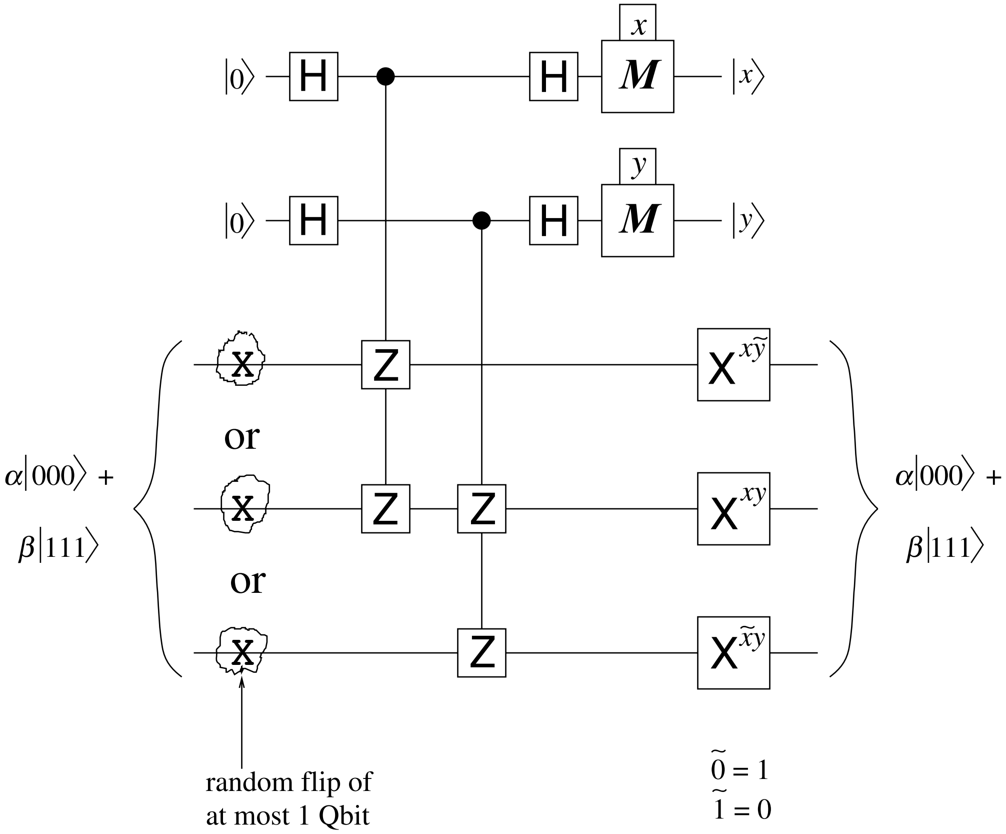


Fig 5.5 An apparently unnecessary complication of the error-correcting circuit in Figure 5.3, which transforms it into the more general form described in Section 5.4. The circuit is equivalent to that in Figure 5.3: (1) the cNOT gates in Figure 5.3 can be replaced by controlled-Z gates if Hadamard gates act on each ancilla before and after the controlled gates act; (2) each of the four controlled-Z gates acts in the same way if its control and target Qbits are interchanged; and (3) pairs of controlled gates with the same control Qbit and two different targets can be combined into a single controlled gate with that control Qbit and a 2-Qbit target operation that is just the product of the two 1-Qbit target operations. The part of the circuit between and including the pairs of Hadamards on the right and left is a simple example of the more complex error-diagnosing circuits that appear in Figures 5.8, 5.9, and N.2 (in Appendix N).

circuits are more elaborate, once we have identified the more general form of quantum errors there are no further conceptual complications in understanding the kinds of procedures that can correct them.

The more general form Qbit errors can assume is discussed in Section 5.3. Somewhat surprisingly, it turns out that the general 1-Qbit error can be viewed as a simple extension of what we have just described: in addition to the possibility of an extraneous $\mathbf{X}$ gate acting on the Qbit, there might also be an extraneous $\mathbf{Z}$ gate or an extraneous $\mathbf{Y} = \mathbf{ZX}$ gate. If we can diagnose and correct for each of these three possible corruptions, then we can correct for arbitrary 1-Qbit errors.

Section 5.4 describes a generalization of the diagnostic scheme we have just exploited for extracting relational information about the Qbits that make up a codeword, by coupling groups of them to ancillary Qbits, which are then measured. It turns out that the operators needed to diagnose the error – generalizations of the operators $\mathbf{Z}_2\mathbf{Z}_1$ and $\mathbf{Z}_1\mathbf{Z}_0$ for the 3-Qbit code – are also useful for defining the more general codewords.

In Section 5.5 we examine two of the most important n-Qbit codes with $n > 3$ that are able to correct general single-Qbit errors: the 5-Qbit and 7-Qbit codes. The 5-Qbit code is the ideal code for general 1-Qbit errors in the same way that the 3-Qbit code is ideal for bit-flip errors. The 7-Qbit code is more likely to be of practical interest, for reasons we shall describe. The earliest quantum error-correcting code – the 9-Qbit code discovered by Shor – is now of only historical interest, and is relegated to Appendix N.

5.3 The physics of error generation

Errors are not, of course, produced by extra gates accidentally appearing in a circuit, as in the oversimplified example of Section 5.2. They are produced by extraneous interactions with the world external to the computer or with computationally irrelevant degrees of freedom of the computer itself. Although one would like the state of the Qbits to evolve only under the action of the unitary transformations imposed by the gates of the computer, inevitably Qbits will interact, even if only weakly, with other physical systems or degrees of freedom having nothing to do with the computation in which the Qbits are participating. In a well-designed computer such spurious interactions will be kept to a minimum, but their disruptive effects on the quantum state of the Qbits can grow continuously from zero, in contrast to disruptive effects on Cbits, which have to exceed a large threshold before a Cbit can change its state. In a quantum computer such spurious changes of state will eventually accumulate to the point where the calculation falls apart, unless ongoing efforts are made to eliminate them.

To characterize the most general way in which a Qbit can be deflected from its computational task, we must finally acknowledge that Qbits are not the only things in the world that are described by quantum states. The quantum theory provides the most fundamental description we have of everything in the world, and it describes everything in the world – not just Qbits – by means of quantum states.

This spectacular expansion of the scope of quantum states might not come as a complete surprise to the nonphysicist reader. I have stressed all along that the quantum state of a Qbit or a collection of Qbits is not a property carried by those Qbits, but a way of concisely summarizing everything we know that has happened to them, to enable us to make statistical predictions about the information we might then be able to extract from them. If quantum states are not properties inherent in the system they describe, but states of the knowledge we have managed to acquire about the prior history of the system – if they somehow incorporate fundamental aspects of how we exchange information with the world outside of us – then they might indeed have an applicability going beyond the particular kinds of systems we have applied them to up until now.

Indeed, nowhere in this exposition of quantum computation has it been necessary to refer to the individual character of the Qbits. Whether they are spinning electrons, polarized photons, atoms in cavities, or any number of other things, the quantum-mechanical description of their computational behavior has been exactly the same. So insofar as the assignment of quantum states to physical systems is a general feature of how we come to grips with the external world, it might not be unreasonable to assign a quantum state $|e\rangle$ to whatever part of the world comes into interactive contact with the Qbit or Qbits – their *environment*. We will not make any specific assumptions about the character of that environment or of the quantum state $|e\rangle$ associated with it, beyond noting that, unlike the state of a single Qbit, the state of the environment is likely to be a state in a space of enormously many dimensions if there is any complexity to the environment that couples, however weakly, to the Qbit.

If, in spite of this recommended point of view, you still feel uncomfortable applying quantum states to noncomputational degrees of freedom, then I invite you to regard $|e\rangle$ as the state of some enormous collection of extra Qbits, from which one would like the computation to be completely decoupled, but which, for reasons beyond our control, somehow manage to interact weakly with the Qbits we are actually interested in. I offer this invitation as a conceptual aid to computer scientists uncomfortable with my claim that quantum states apply to the description of arbitrary physical systems. But I also note that in recent years a few physicists have suggested that the entire world should indeed be viewed as an enormous collection of Qbits – a position that has not attracted many adherents to date.

Returning from grand world views to the practical reality of errors in a quantum computation, we shall regard a single Qbit, initially in the state $|x\rangle$ ($x = 0$ or 1), as being part of a larger system consisting of the Qbit plus its environment, initially in the state $|e\rangle|x\rangle$. In the ideal case, as the Qbit evolves under 1-Qbit unitary gates or interacts with other Qbits under 2-Qbit unitary gates, it stays unentangled with its environment. The environmental component of the state is then irrelevant to the computational process and can be ignored, as we have been doing up to now.

Unfortunately, however, interactions with the environment will in general transform and entangle the states of the Qbit and its environment. The most general way in which this can come about can be expressed in the form

$$\begin{aligned} |e\rangle|0\rangle &\to |e_0\rangle|0\rangle + |e_1\rangle|1\rangle, \\ |e\rangle|1\rangle &\to |e_2\rangle|0\rangle + |e_3\rangle|1\rangle, \end{aligned} \tag{5.7}$$

where $|e\rangle$ is the initially uncorrelated state of the environment and $|e_0\rangle, \ldots, |e_3\rangle$ are possible final environmental states. The environmental final states are not necessarily orthogonal or normalized, and are constrained only by the requirement that the two states on the right side of (5.7) should be orthogonal, since the Qbit–environment interaction is required, like any other physical interaction, to lead to a unitary development in time. This corruption of a computation by the entanglement of the state of Qbits with the state of their environment is called *decoherence*. It is the primary enemy of quantum computation.

Included in (5.7) are cases like the oversimplified one we examined in Section 5.2, in which the Qbit remains isolated from the environment ($|e_i\rangle = a_i|e\rangle,\ i = 0, \ldots, 3$) but still suffers in that isolation an unintended unitary evolution. But (5.7) also includes the case of major practical interest. This is the case in which the interaction with the environment has a small but otherwise quite general entangling effect on the Qbit:

$$|e_0\rangle \approx |e_3\rangle \approx |e\rangle; \qquad \langle e_1|e_1\rangle,\ \ \langle e_2|e_2\rangle \ll 1. \tag{5.8}$$

In dealing with such entangling interactions with the environment, it is useful to introduce projection operators

$$\mathbf{P}_x = \frac{1 + (-1)^x \mathbf{Z}}{2}, \tag{5.9}$$

which project onto the 1-Qbit states $|x\rangle,\ \ x = 0, 1$. Using these projection operators, we can combine the two time evolutions in (5.7) into a single form:

$$|e\rangle|x\rangle \to \big(\big[|e_0\rangle\mathbf{1} + |e_1\rangle\mathbf{X}\big]\mathbf{P}_0\big)|x\rangle + \big(\big[|e_2\rangle\mathbf{X} + |e_3\rangle\mathbf{1}\big]\mathbf{P}_1\big)|x\rangle. \tag{5.10}$$

In (5.10) I have introduced the convenient notation $|e\rangle\mathbf{U}$ to describe the linear operator from a 1-Qbit to a many-Qbit space that takes the 1-Qbit state $|\psi\rangle$ into the many-Qbit state $|e\rangle \otimes \mathbf{U}|\psi\rangle$; like most embellishments of Dirac notation it is defined so that the appropriate form of the associative law holds:

$$\big(|e\rangle\mathbf{U}\big)|\psi\rangle = |e\rangle \otimes \mathbf{U}|\psi\rangle. \tag{5.11}$$

Using the explicit form (5.9) of the two projection operators, defining[1]

$$\mathbf{Y} = \mathbf{Z}\mathbf{X}, \tag{5.12}$$

and continuing to use the notational convention (5.11), we can rewrite (5.10) as

$$|e\rangle|x\rangle \to \biggl(\frac{|e_0\rangle + |e_3\rangle}{2}\mathbf{1} + \frac{|e_0\rangle - |e_3\rangle}{2}\mathbf{Z} \\ + \frac{|e_2\rangle + |e_1\rangle}{2}\mathbf{X} + \frac{|e_2\rangle - |e_1\rangle}{2}\mathbf{Y}\biggr)|x\rangle. \tag{5.13}$$

There is nothing special about the particular environmental states appearing in (5.13), so we can rewrite it more compactly in terms of four other (in general neither normalized nor orthogonal) states $|a\rangle$, $|b\rangle$, $|c\rangle$, and $|d\rangle$ of the environment as

$$|e\rangle|x\rangle \to \bigl(|d\rangle\mathbf{1} + |a\rangle\mathbf{X} + |b\rangle\mathbf{Y} + |c\rangle\mathbf{Z}\bigr)|x\rangle. \tag{5.14}$$

The time development represented by the arrow in (5.14) is unitary and therefore linear, so the combination of environmental states and unitary operators on the right side of (5.14) acts linearly on $|x\rangle$. Therefore (5.14) holds not only for $|e\rangle|0\rangle$ and $|e\rangle|1\rangle$ but also for any superposition $\alpha|e\rangle|0\rangle + \beta|e\rangle|1\rangle = |e\rangle\bigl(\alpha|0\rangle + \beta|1\rangle\bigr) = |e\rangle|\psi\rangle$, in the form

$$|e\rangle|\psi\rangle \to \bigl(|d\rangle\mathbf{1} + |a\rangle\mathbf{X} + |b\rangle\mathbf{Y} + |c\rangle\mathbf{Z}\bigr)|\psi\rangle. \tag{5.15}$$

The actions of **X**, **Z**, and **Y** on the uncorrupted state $|\psi\rangle$ are sometimes described as subjecting the Qbit to a bit-flip error, a phase error, and a combined bit-flip and phase error. Using this terminology, a general environmental degradation of the state of a Qbit, which can always be put in the form (5.15), can be viewed as a superposition of no error (**1**), a bit-flip error (**X**), a combined bit-flip and phase error (**Y**), and a phase error (**Z**). The oversimplified example of Section 5.2 ignored the possibility of phase errors (**Z**) and combined errors (**Y**).

If we were to extend this analysis to the corruption of an n-Qbit codeword $|\Psi\rangle_n$, we would end up with a combined state of the codeword and the environment of the form

$$|e\rangle|\Psi\rangle \to \sum_{\mu_1=0}^{3} \cdots \sum_{\mu_n=0}^{3} |e_{\mu_1\cdots\mu_n}\rangle \mathbf{X}^{(\mu_1)} \otimes \cdots \otimes \mathbf{X}^{(\mu_n)}|\Psi\rangle, \tag{5.16}$$

where

$$\mathbf{X}^{(0)} = \mathbf{1}, \qquad \mathbf{X}^{(1)} = \mathbf{X}, \qquad \mathbf{X}^{(2)} = \mathbf{Y}, \qquad \mathbf{X}^{(3)} = \mathbf{Z}. \tag{5.17}$$

The construction of error-correcting codewords requires a physical assumption analogous to the assumption in Section 5.2 that at most a

[1] This **Y** differs by a factor of i from the **Y** briefly used in Section 1.4.

single Qbit in a codeword suffers a flip error. If $|\Psi\rangle$ is a state of a small number n of Qbits that make up an n-Qbit codeword, then the probability of corruption of the codeword is so small that the terms in (5.16) differing from the term $\mathbf{1}\otimes\cdots\otimes\mathbf{1}$ that reproduces the uncorrupted state are dominated by those in which only a single one of the $\mathbf{X}^{(\mu_i)}$ differs from $\mathbf{1}$. If this condition is met, then the general form (5.16) of a corrupted n-Qbit codeword is a superposition of terms in which each individual Qbit making up the word has suffered a degradation of the form (5.15):

$$|e\rangle|\Psi\rangle \to \left(|d\rangle\mathbf{1} + \sum_{i=0}^{n-1} |a_i\rangle\mathbf{X}_i + |b_i\rangle\mathbf{Y}_i + |c_i\rangle\mathbf{Z}_i\right)|\Psi\rangle. \tag{5.18}$$

One can allow for the more general possibility of two or more Qbits in a codeword being corrupted together if one is willing to use longer codewords to correct such errors. The examples of error correction given below are all at the level of single-Qbit errors of the form (5.18) in the codeword. The extent to which the dominant sources of error will actually be of this form may well depend on the kind of physical system used to realize the Qbits. Eventually the theory of quantum error correction will have to face this issue. Meanwhile this possible future source of difficulty should not distract you from appreciating how remarkable it is that an error-correction procedure exists at all, even in the restricted setting of single-Qbit errors.

To correct 1-Qbit errors we require a procedure that restores a corrupted state of the form

$$|d\rangle|\Psi\rangle + \sum_{i=0}^{n-1}\big(|a_i\rangle\mathbf{X}_i|\Psi\rangle + |b_i\rangle\mathbf{Y}_i|\Psi\rangle + |c_i\rangle\mathbf{Z}_i|\Psi\rangle\big) \tag{5.19}$$

to the uncorrupted form

$$|e\rangle|\Psi\rangle, \tag{5.20}$$

where $|e\rangle$ is the environmental state accompanying whichever of the $3n+1$ terms in (5.19) our error-correction procedure has projected the corrupted state onto. If the term in $\mathbf{X}_i$ were the only one present in (5.19), we could use a 3-Qbit codeword and achieve this projection by applying precisely the error-correction technique described in Section 5.2. But to deal with the additional possibilities associated with the terms in $\mathbf{Y}_i$ and $\mathbf{Z}_i$ we require longer codewords and more elaborate diagnostic methods.

5.4 Diagnosing error syndromes

Before turning to specific quantum error-correcting codes, it is useful to anticipate the general structure of the gates we will be using to identify and project onto a particular term in the general 1-Qbit corruption (5.19) of a codeword. As noted earlier, these will be generalizations of the controlled $\mathbf{Z}_2\mathbf{Z}_1$ and $\mathbf{Z}_1\mathbf{Z}_0$ gates used to diagnose errors in the artificial case in which only bit-flip errors are allowed.

Let $\mathbf{A}$ be *any* n-Qbit Hermitian operator whose square is the unit operator:

$$\mathbf{A}^2 = \mathbf{1}. \tag{5.21}$$

It follows from (5.21) that $\mathbf{A}$ is unitary, since $\mathbf{A}^\dagger = \mathbf{A}$. The eigenvalues of $\mathbf{A}$ can only be 1 or -1, since $\mathbf{A}$ acting twice on an eigenstate must act as the identity $\mathbf{1}$. The projection operators onto the subspaces of states with eigenvalue $+1$ and -1 are, respectively,

$$\mathbf{P}_0^A = \frac{\mathbf{1}+\mathbf{A}}{2} \qquad \text{and} \qquad \mathbf{P}_1^A = \frac{\mathbf{1}-\mathbf{A}}{2}. \tag{5.22}$$

Since $\mathbf{P}_0 + \mathbf{P}_1 = \mathbf{1}$, any state $|\psi\rangle$ can be expressed as a superposition of its projections onto these two subspaces: $|\psi\rangle = \mathbf{P}_0|\psi\rangle + \mathbf{P}_1|\psi\rangle$.

The operators $\mathbf{Z}_2\mathbf{Z}_1$ and $\mathbf{Z}_1\mathbf{Z}_0$ encountered in the 3-Qbit code for correcting bit-flip errors are examples of such $\mathbf{A}$. In the more general cases we shall be examining, the operators $\mathbf{A}$ will be more general products of both $\mathbf{Z}$ and $\mathbf{X}$ operators associated with different Qbits in the codeword; for example $\mathbf{A} = \mathbf{Z}_4\mathbf{X}_3\mathbf{Z}_2\mathbf{X}_1\mathbf{X}_0$.

In addition to the n Qbits on which $\mathbf{A}$ acts, we introduce an ancillary Qbit and consider the controlled operator $\mathbf{C}^A$, which we write here in the alternative form $\mathbf{cA}$ to avoid having subscripts on superscripts, which acts as $\mathbf{A}$ on the n Qbits when the state of the ancilla is $|1\rangle$ and as the identity when the state of the ancilla is $|0\rangle$. If the state of the ancilla is a superposition of $|0\rangle$ and $|1\rangle$, the action of $\mathbf{cA}$ is defined by linearity. When $\mathbf{A}$ is a product of 1-Qbit operators, the operator $\mathbf{cA}$ can be taken to be a product of ordinary 2-Qbit controlled operators. If $\mathbf{A} = \mathbf{Z}_4\mathbf{X}_3\mathbf{Z}_2\mathbf{X}_1\mathbf{X}_0$, then $\mathbf{cA}$ would be $\mathbf{cZ}_4\mathbf{cX}_3\mathbf{cZ}_2\mathbf{cX}_1\mathbf{cX}_0$, where each of the five terms has a different target Qbit, but all are controlled by one and the same ancilla.

If the ancilla is initially in the state $|0\rangle$ and one applies a Hadamard transform $\mathbf{H}$ to the ancilla both before and after applying $\mathbf{cA}$ to the $n+1$ Qbits and the initial state of the n Qbits is $|\Psi\rangle$, then the n Qbits will end up entangled with the ancilla in the state

$$\begin{aligned}
\big(\mathbf{H}\otimes\mathbf{1}\big)\mathbf{cA}\big(\mathbf{H}\otimes\mathbf{1}\big)|0\rangle|\Psi\rangle &= \big(\mathbf{H}\otimes\mathbf{1}\big)\mathbf{cA}\tfrac{1}{\sqrt{2}}\big(|0\rangle+|1\rangle\big)|\Psi\rangle \\
&= \big(\mathbf{H}\otimes\mathbf{1}\big)\tfrac{1}{\sqrt{2}}\big(|0\rangle|\Psi\rangle+|1\rangle\mathbf{A}|\Psi\rangle\big) \\
&= \tfrac{1}{2}\big(|0\rangle+|1\rangle\big)|\Psi\rangle+\tfrac{1}{2}\big(|0\rangle-|1\rangle\big)\mathbf{A}|\Psi\rangle \\
&= |0\rangle\tfrac{1}{2}\big(\mathbf{1}+\mathbf{A}\big)|\Psi\rangle+|1\rangle\tfrac{1}{2}\big(\mathbf{1}-\mathbf{A}\big)|\Psi\rangle \\
&= |0\rangle\mathbf{P}_0^A|\Psi\rangle+|1\rangle\mathbf{P}_1^A|\Psi\rangle.
\end{aligned} \tag{5.23}$$

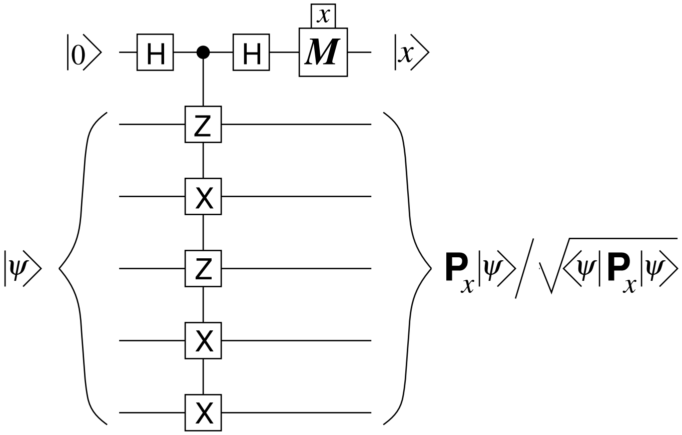

$|0\rangle$
H
H
x
$\boldsymbol{M}$
$|x\rangle$
Z
X
Z
X
X
$|\psi\rangle$
$\mathbf{P}_x|\psi\rangle \Big/ \sqrt{\langle\psi|\mathbf{P}_x|\psi\rangle}$

Fig 5.6 The way in which measurement gates are employed in quantum error correction. The ancilla (upper wire) is initially in the state zero. The remaining five Qbits are initially in the state $|\psi\rangle$. If the measurement gate acting on the ancilla gives the result x (0 or 1) then the final state of the five Qbits will be the (renormalized) projection $\mathbf{P}_x|\psi\rangle$ of the initial state into the subspace spanned by the eigenstates of $\mathbf{Z}_4\mathbf{X}_3\mathbf{Z}_2\mathbf{X}_1\mathbf{X}_0$ with eigenvalue $(-1)^x$.

If we now measure the ancilla, then according to the generalized Born rule, if the measurement gate indicates 0 or 1, then the state of the n Qbits becomes the (renormalized) projection of $|\Psi\rangle$ into the subspace of positive (eigenvalue $+1$) or negative (eigenvalue -1) eigenstates of $\mathbf{A}$. This is illustrated for the case $\mathbf{A} = \mathbf{Z}_4\mathbf{X}_3\mathbf{Z}_2\mathbf{X}_1\mathbf{X}_0$ in Figure 5.6.

This procedure is called *measuring* $\mathbf{A}$ or a *measurement* of $\mathbf{A}$. The terminology reflects the fact that it is a generalization of the ordinary process of measuring a single Qbit, to which it reduces when $n = 1$ and $\mathbf{A} = \mathbf{Z}$. In that case the subspaces spanned by the positive and negative eigenstates of $\mathbf{Z}$ are just the one-dimensional subspaces spanned by $|0\rangle$ and $|1\rangle$, and the probabilities of the two outcomes, as one can easily check, are indeed given by the Born rule.

In error correction one needs several such Hermitian operators, each squaring to unity, all acting on the same n Qbits. For concreteness consider the case of three such operators, $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$. Introduce an ancillary Qbit for each operator, labeling the ancillas 0, 1, and 2, and introduce controlled operators $\mathbf{cA}$, $\mathbf{cB}$, and $\mathbf{cC}$, each controlled by the corresponding ancilla. Now apply Hadamards to each of the ancillas (each initially taken to be in the state $|0\rangle$), both before and after the product of all the controlled operators acts. The result (see Figure 5.7) is the obvious generalization of (5.23), taking $|0\rangle|0\rangle|0\rangle|\Psi\rangle$ into

$$\begin{aligned}
&\big(\mathbf{H}_2\mathbf{H}_1\mathbf{H}_0\big)\big(\mathbf{cCcBcA}\big)\big(\mathbf{H}_2\mathbf{H}_1\mathbf{H}_0\big)|0\rangle|0\rangle|0\rangle|\Psi\rangle \\
&\quad = \sum_{x_2=0}^{1}\sum_{x_1=0}^{1}\sum_{x_0=0}^{1} |x_2\rangle|x_1\rangle|x_0\rangle \left(\frac{1+(-1)^{x_2}\mathbf{C}}{2}\right)\left(\frac{1+(-1)^{x_1}\mathbf{B}}{2}\right) \\
&\qquad\qquad \times \left(\frac{1+(-1)^{x_0}\mathbf{A}}{2}\right)|\Psi\rangle \\
&\quad = \sum_{x_2=0}^{1}\sum_{x_1=0}^{1}\sum_{x_0=0}^{1} |x_2\rangle|x_1\rangle|x_0\rangle \mathbf{P}^{C}_{x_2}\mathbf{P}^{B}_{x_1}\mathbf{P}^{A}_{x_0}|\Psi\rangle. \qquad (5.24)
\end{aligned}$$

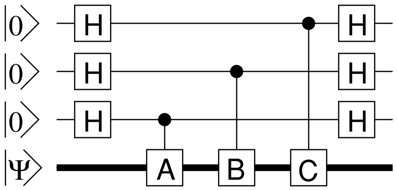


Fig 5.7 **A**, **B**, and **C** are commuting operators satisfying $\mathbf{A}^2 = \mathbf{B}^2 = \mathbf{C}^2 = \mathbf{1}$. They act on the n-Qbit state $|\Psi\rangle$ associated with the thick lower wire. The effect of measuring the three ancillas (top three wires) is to project the state of the n Qbits associated with the lower wire into its component in one of the eight eigenspaces of **A**, **B**, and **C**. If the results of measuring the control bits associated with **A**, **B**, and **C** are x_0, x_1, and x_2 then the projection is into the eigenspace with eigenvalues $(-1)^{x_0}$, $(-1)^{x_1}$, and $(-1)^{x_2}$. Such a process is called "measuring **A**, **B**, and **C**." When $n = 3$ and **A**, **B**, and **C** are three different 1-Qbit **Z** operators, the process is equivalent to an ordinary measurement of the three Qbits on which the three **Z** operators act.

If **A**, **B**, and **C** all *commute* – which is always the case in the examples relevant to error correction – then the state

$$\mathbf{P}^C_{x_2}\mathbf{P}^B_{x_1}\mathbf{P}^A_{x_0}|\Psi\rangle = \left(\frac{1+(-1)^{x_2}\mathbf{C}}{2}\right)\left(\frac{1+(-1)^{x_1}\mathbf{B}}{2}\right)\left(\frac{1+(-1)^{x_0}\mathbf{A}}{2}\right) \tag{5.25}$$

is an eigenstate of all the operators **C**, **B**, and **A**, with respective eigenvalues

$$(-1)^{x_2}, (-1)^{x_1}, \text{ and } (-1)^{x_0}. \tag{5.26}$$

This follows directly from the fact that if $\mathbf{V}^2 = 1$ then

$$\mathbf{V}\left(\frac{1+(-1)^x\mathbf{V}}{2}\right) = (-1)^x\left(\frac{1+(-1)^x\mathbf{V}}{2}\right). \tag{5.27}$$

So measurement of the three ancillas projects the n Qbits into one of the eight simultaneous eigenspaces of the three commuting operators **C**, **B**, and **A**, and the outcome $x_2x_1x_0$ of the measurement determines which eigenspace it is. This process is described as a joint measurement of **C**, **B**, and **A**.

Note that if **A**, **B**, and **C** are 1-Qbit operators $\mathbf{Z}_i$, $\mathbf{Z}_j$, and $\mathbf{Z}_k$ that act on the ith, jth, and kth of the n Qbits, then this process reduces to the ordinary measurement of those three Qbits, since $\frac{1}{2}(1+(-1)^x\mathbf{Z})$ projects onto the 1-Qbit state $|x\rangle$. The two equivalent error-correction circuits in Figures 5.3 and 5.5 are measurements, in this generalized sense, of the two commuting operators $\mathbf{A} = \mathbf{Z}_2\mathbf{Z}_1$ and $\mathbf{B} = \mathbf{Z}_1\mathbf{Z}_0$.

The form (5.18) of a general 1-Qbit error on an n-Qbit codeword reveals that to correct errors it is necessary to make a measurement, in this more general sense of the term, that projects a possibly corrupted codeword into an identifiable one of $1 + 3n$ orthogonal two-dimensional spaces: one two-dimensional subspace for the uncorrupted codeword $|\Psi\rangle$, and $3n$ additional two-dimensional subspaces for each of the 1-Qbit error terms $\mathbf{X}_i|\Psi\rangle$, $\mathbf{Y}_i|\Psi\rangle$, and $\mathbf{Z}_i|\Psi\rangle$, $i = 0, \ldots, n-1$, in (5.18). Thus the 2^n-dimensional space spanned by all the states of the n Qbits must be large enough to contain $1 + 3n$ orthogonal two-dimensional subspaces, giving us the condition

$$2^{n-1} \geq 3n + 1 \tag{5.28}$$

on an n-Qbit code capable of correcting a general 1-Qbit error. The lowest n satisfying this condition is $n = 5$, for which it holds as an equality. Remarkably, there is indeed a 5-Qbit code for which this can be done. This is reminiscent of the situation in Section 5.2, where it was necessary only to discriminate between the uncorrupted codeword $|\Psi\rangle$ and the n NOT-corruptions $\mathbf{X}_i|\Psi\rangle$. There the number of Qbits had to satisfy (5.5), which is first satisfied (again as an equality) when $n = 3$.

The 5-Qbit code is the most compact and elegant of the quantum error-correcting codes, but it suffers from the fact that it is difficult to construct the appropriate generalizations of 1- and 2-Qbit gates between codewords. I therefore go on to describe a second, 7-Qbit code, which overcomes this problem. The first quantum error-correcting code, discovered by Peter Shor, which uses a 9-Qbit generalization of the 3-Qbit code of Section 5.2, is now of solely historical interest. It is described in Appendix N.

5.5 The 5-Qbit error-correcting code

The two 5-Qbit code words $|\overline{0}\rangle$ and $|\overline{1}\rangle$ are most conveniently defined in terms of the very operators, described in general terms in Section 5.4, that will be used to diagnose the error syndrome. So we begin by specifying those operators.

To distinguish $1 + (3 \times 5) = 16$ mutually orthogonal two-dimensional subspaces we require four such mutually commuting Hermitian operators that square to unity, since each can independently have two eigenvalues (± 1) and $2^4 = 16$. These operators are defined as follows:

$$\begin{aligned}
\mathbf{M}_0 &= \mathbf{Z}_1\mathbf{X}_2\mathbf{X}_3\mathbf{Z}_4,\\
\mathbf{M}_1 &= \mathbf{Z}_2\mathbf{X}_3\mathbf{X}_4\mathbf{Z}_0,\\
\mathbf{M}_2 &= \mathbf{Z}_3\mathbf{X}_4\mathbf{X}_0\mathbf{Z}_1,\\
\mathbf{M}_3 &= \mathbf{Z}_4\mathbf{X}_0\mathbf{X}_1\mathbf{Z}_2.
\end{aligned} \tag{5.29}$$

Each of the $\mathbf{M}_i$ squares to unity because each is a product of commuting operators that square to unity. To check that the $\mathbf{M}_i$ are mutually commuting, note that all the individual $\mathbf{X}_i$ and $\mathbf{Z}_j$ operators commute with one another except for an $\mathbf{X}_i$ and $\mathbf{Z}_i$ with the same index, which anticommute: $\mathbf{X}_i\mathbf{Z}_i = -\mathbf{Z}_i\mathbf{X}_i$. But in converting the product of any two different $\mathbf{M}_i$ to the product in the reverse order by reversing the orders of the individual $\mathbf{X}_i$ and $\mathbf{Z}_i$ operators that make them up, one always encounters exactly two interchanges that result in a minus sign.

One might be tempted to break the irritating asymmetry of (5.30) by adding to the list

$$\mathbf{M}_4 = \mathbf{Z}_0\mathbf{X}_1\mathbf{X}_2\mathbf{Z}_3, \tag{5.30}$$

but it is not independent of the other four. Every X_i and Z_i appears exactly twice in the product of all five M_i, so the product must be either 1 or -1. One easily checks that

$$\mathsf{M}_0\mathsf{M}_1\mathsf{M}_2\mathsf{M}_3\mathsf{M}_4 = \mathbf{1}, \tag{5.31}$$

and therefore

$$\mathsf{M}_4 = \mathsf{M}_0\mathsf{M}_1\mathsf{M}_2\mathsf{M}_3. \tag{5.32}$$

The 5-Qbit codewords are most clearly and usefully defined in terms of the M_i (rather than writing out their lengthy explicit expansions in computational-basis states):

$$\begin{aligned} |\overline{0}\rangle &= \tfrac{1}{4}(\mathbf{1}+\mathsf{M}_0)(\mathbf{1}+\mathsf{M}_1)(\mathbf{1}+\mathsf{M}_2)(\mathbf{1}+\mathsf{M}_3)|00000\rangle, \\ |\overline{1}\rangle &= \tfrac{1}{4}(\mathbf{1}+\mathsf{M}_0)(\mathbf{1}+\mathsf{M}_1)(\mathbf{1}+\mathsf{M}_2)(\mathbf{1}+\mathsf{M}_3)|11111\rangle. \end{aligned} \tag{5.33}$$

Before examining how one might produce five Qbits in either of these states, we discuss how the states work to correct 1-Qbit errors.

Since each M flips two Qbits, $|\overline{0}\rangle$ is a superposition of computational-basis states with an odd number of zeros (and an even number of ones), while $|\overline{1}\rangle$ is a superposition of states with an odd number of ones (and an even number of zeros). Consequently the two codeword states are orthogonal. They are also normalized to unity. Since $\mathsf{M}_i^2 = 1$,

$$(\mathbf{1}+\mathsf{M}_i)^2 = 2(\mathbf{1}+\mathsf{M}_i). \tag{5.34}$$

So we have

$$\begin{aligned} \langle\overline{0}|\overline{0}\rangle &= \langle 00000|(\mathbf{1}+\mathsf{M}_0)(\mathbf{1}+\mathsf{M}_1)(\mathbf{1}+\mathsf{M}_2)(\mathbf{1}+\mathsf{M}_3)|00000\rangle, \\ \langle\overline{1}|\overline{1}\rangle &= \langle 11111|(\mathbf{1}+\mathsf{M}_0)(\mathbf{1}+\mathsf{M}_1)(\mathbf{1}+\mathsf{M}_2)(\mathbf{1}+\mathsf{M}_3)|11111\rangle. \end{aligned} \tag{5.35}$$

If we expand the products of $\mathbf{1}+\mathsf{M}_i$ into 16 terms, the term $\mathbf{1}$ contributes 1 to $\langle\overline{0}|\overline{0}\rangle$ and to $\langle\overline{1}|\overline{1}\rangle$. Each of the remaining 15 terms can be reduced, using (5.31) (and the fact that each $\mathsf{M}_i^2 = \mathbf{1}$), to either a single M_i or a product of two ($i = 0, \ldots, 4$). So each of the 15 terms flips either two or four Qbits and contributes 0 to the inner products.

Because the M_i all commute and because

$$\mathsf{M}_i(\mathbf{1}+\mathsf{M}_i) = \mathbf{1}+\mathsf{M}_i, \tag{5.36}$$

the states $|\overline{0}\rangle$, $|\overline{1}\rangle$, and their superpositions

$$|\Psi\rangle = \alpha|\overline{0}\rangle + \beta|\overline{1}\rangle \tag{5.37}$$

are all eigenstates of each of the M_i with eigenvalue 1.

The 15 possible corruptions of (5.37) appearing in the corrupted state (5.18) are also eigenstates of the M_i, distinguished by the

Table 5.2. The four error-syndrome operators $\mathbf{M}_i$ for the 5-Qbit code, and whether each of them commutes (+) or anticommutes (−) with each of the 15 operators $\mathbf{X}_i$, $\mathbf{Y}_i$, and $\mathbf{Z}_i$, $i = 1, \ldots, 5$, associated with the 15 different terms in the corrupted codeword. Note that each of the 15 columns, and the 16th column associated with **1** (no error), has a unique pattern of + and − signs.

	$\mathbf{X}_0\mathbf{Y}_0\mathbf{Z}_0$	$\mathbf{X}_1\mathbf{Y}_1\mathbf{Z}_1$	$\mathbf{X}_2\mathbf{Y}_2\mathbf{Z}_2$	$\mathbf{X}_3\mathbf{Y}_3\mathbf{Z}_3$	$\mathbf{X}_4\mathbf{Y}_4\mathbf{Z}_4$	$\mathbf{1}$
$\mathbf{M}_0 = \mathbf{Z}_1\mathbf{X}_2\mathbf{X}_3\mathbf{Z}_4$	$+++$	$--+$	$+--$	$+--$	$--+$	$+$
$\mathbf{M}_1 = \mathbf{Z}_2\mathbf{X}_3\mathbf{X}_4\mathbf{Z}_0$	$--+$	$+++$	$--+$	$+--$	$+--$	$+$
$\mathbf{M}_2 = \mathbf{Z}_3\mathbf{X}_4\mathbf{X}_0\mathbf{Z}_1$	$+--$	$--+$	$+++$	$--+$	$+--$	$+$
$\mathbf{M}_3 = \mathbf{Z}_4\mathbf{X}_0\mathbf{X}_1\mathbf{Z}_2$	$+--$	$+--$	$--+$	$+++$	$--+$	$+$

$15 = 2^4 - 1$ other possible sets of eigenvalues ±1 that the four $\mathbf{M}_i$ ($i = 0, \ldots, 3$) can have. To see this, note first that each $\mathbf{X}_i$, $\mathbf{Y}_i$, and $\mathbf{Z}_i$ commutes or anticommutes with all four $\mathbf{M}_i$. Therefore each of the terms $\mathbf{X}_i|\Psi\rangle$, $\mathbf{Y}_i|\Psi\rangle$, and $\mathbf{Z}_i|\Psi\rangle$ appearing in (5.18) is indeed an eigenstate of each $\mathbf{M}_i$ with eigenvalue 1 or −1.

Table 5.2 indicates whether each $\mathbf{M}_i$ commutes (+) or anticommutes (−) with each of the $\mathbf{X}_i$, $\mathbf{Y}_i$, $\mathbf{Z}_i$, and (trivially) the unit operator **1**. Inspection of the table reveals that each of the 16 possible binary columns of four symbols (+ or −) appears in exactly one column. Therefore, when the four $\mathbf{M}_i$ are measured, the corrupted state (5.18) is projected back to its original form if all four eigenvalues are +1, or projected onto one of the 15 corrupted states $\mathbf{X}_0|\Psi\rangle, \ldots, \mathbf{Z}_4|\Psi\rangle$ depending on which column in the table describes the eigenvalues. In each corrupted case the original state can be restored by application of the corresponding unitary transformation $\mathbf{X}_i$, $-\mathbf{Y}_i = \mathbf{X}_i\mathbf{Z}_i$, or $\mathbf{Z}_i$ to the appropriate Qbit. A circuit that measures the four operators (5.29) is shown in Figure 5.8.

The perfect efficiency of the 5-Qbit code leads to a straightforward way to manufacture the two 5-Qbit codeword states (5.33). As noted above, the 16 distinct sets of eigenvalues for the four mutually commuting operators $\mathbf{M}_i$ decompose the 32-dimensional space of five Qbits into 16 mutually orthogonal two-dimensional subspaces, spanned by $|\overline{0}\rangle$ and $|\overline{1}\rangle$ and by each of their 15 pairs of 1-Qbit corruptions.

The two-fold degeneracy of the four $\mathbf{M}_i$ within each of these 16 subspaces is lifted by the operator

$$\overline{\mathbf{Z}} = \mathbf{Z}_0\mathbf{Z}_1\mathbf{Z}_2\mathbf{Z}_3\mathbf{Z}_4, \tag{5.38}$$

which commutes with all the $\mathbf{M}_i$. Since $|00000\rangle$ and $|11111\rangle$ are eigenstates of $\overline{\mathbf{Z}}$ with eigenvalues 1 and −1, and since $\overline{\mathbf{Z}}$ commutes with $\mathbf{Z}_i$,

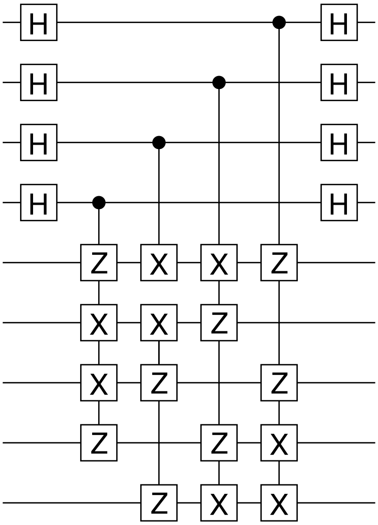


Fig 5.8 A circuit to measure the error syndrome for the 5-Qbit code. The five Qbits are the five lower wires. The four upper wires are the ancillas to be measured in the manner of Figure 5.7, associated with measuring the four commuting operators $\mathbf{Z}_1\mathbf{X}_2\mathbf{X}_3\mathbf{Z}_4$, $\mathbf{Z}_2\mathbf{X}_3\mathbf{X}_4\mathbf{Z}_0$, $\mathbf{Z}_3\mathbf{X}_4\mathbf{X}_0\mathbf{Z}_1$, and $\mathbf{Z}_4\mathbf{X}_0\mathbf{X}_1\mathbf{Z}_2$ of (5.29). When controlled-Z gates are present together with controlled-NOT gates, the figure is more readable if the cNOT gates are represented as controlled-X gates.

while anticommuting with $\mathbf{X}_i$ and $\mathbf{Y}_i$, it follows that

$$\begin{aligned} \overline{\mathbf{Z}}|\overline{0}\rangle &= |\overline{0}\rangle, & \overline{\mathbf{Z}}|\overline{1}\rangle &= -|\overline{1}\rangle, \\ \overline{\mathbf{Z}}\mathbf{Z}_i|\overline{0}\rangle &= \mathbf{Z}_i|\overline{0}\rangle, & \overline{\mathbf{Z}}\mathbf{Z}_i|\overline{1}\rangle &= -\mathbf{Z}_i|\overline{1}\rangle, \\ \overline{\mathbf{Z}}\mathbf{X}_i|\overline{0}\rangle &= -\mathbf{X}_i|\overline{0}\rangle, & \overline{\mathbf{Z}}\mathbf{X}_i|\overline{1}\rangle &= \mathbf{X}_i|\overline{1}\rangle, \\ \overline{\mathbf{Z}}\mathbf{Y}_i|\overline{0}\rangle &= -\mathbf{Y}_i|\overline{0}\rangle, & \overline{\mathbf{Z}}\mathbf{Y}_i|\overline{1}\rangle &= \mathbf{Y}_i|\overline{1}\rangle. \end{aligned} \tag{5.39}$$

Consequently if one takes five Qbits in any state you like (perhaps most conveniently $|00000\rangle$) and measures the four $\mathbf{M}_i$ together with $\overline{\mathbf{Z}}$, one projects the Qbits into one of the 32 states

$$|\overline{0}\rangle,\quad \mathbf{X}_i|\overline{0}\rangle,\quad \mathbf{Y}_i|\overline{0}\rangle,\quad \mathbf{Z}_i|\overline{0}\rangle,\quad |\overline{1}\rangle,\quad \mathbf{X}_i|\overline{1}\rangle,\quad \mathbf{Y}_i|\overline{1}\rangle,\quad \mathbf{Z}_i|\overline{1}\rangle, \tag{5.40}$$

and learns from the results of the measurement which it is. Just as in the error-correction procedure, if the state is not $|\overline{0}\rangle$ or $|\overline{1}\rangle$ we can restore it to either of these forms by applying the appropriate $\mathbf{X}_i$, $\mathbf{Y}_i$, or $\mathbf{Z}_i$. If we wish to initialize the five Qbits to $|\overline{0}\rangle$ we can apply $\overline{\mathbf{X}}$, where

$$\overline{\mathbf{X}} = \mathbf{X}_0\mathbf{X}_1\mathbf{X}_2\mathbf{X}_3\mathbf{X}_4, \tag{5.41}$$

should the measurement indicate that the error-corrected state is $|\overline{1}\rangle$. This process of using a generalized measurement to produce five Qbits in the state $|\overline{0}\rangle$ is analogous to the procedure of using an ordinary measurement to produce a single Qbit in the state $|0\rangle$ described in Section 1.10.

There is quite a different way to construct the 5-Qbit codewords, by applying a set of 1- and 2-Qbit unitary gates to an uncoded 1-Qbit state and four ancillary Qbits all initially in the state $|0\rangle$. This is described in Section 5.9.

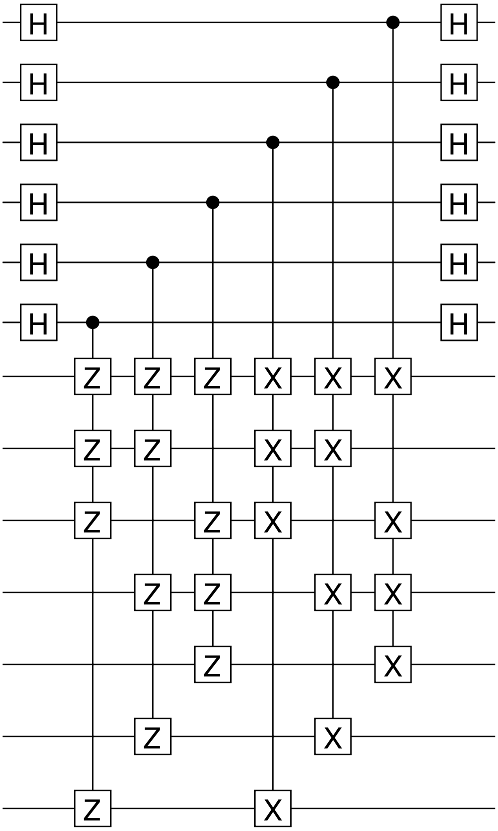


Fig 5.9 A circuit to measure the error syndrome for the 7-Qbit code. The seven Qbits are the seven lower wires. The six upper wires are the ancillas to be measured, resulting in a measurement of the six commuting operators $\mathbf{Z}_0\mathbf{Z}_4\mathbf{Z}_5\mathbf{Z}_6$, $\mathbf{Z}_1\mathbf{Z}_3\mathbf{Z}_5\mathbf{Z}_6$, $\mathbf{Z}_2\mathbf{Z}_3\mathbf{Z}_4\mathbf{Z}_6$, $\mathbf{X}_0\mathbf{X}_4\mathbf{X}_5\mathbf{X}_6$, $\mathbf{X}_1\mathbf{X}_3\mathbf{X}_5\mathbf{X}_6$, and $\mathbf{X}_2\mathbf{X}_3\mathbf{X}_4\mathbf{X}_6$ of (5.42).

5.6 The 7-Qbit error-correcting code

The 5-Qbit code is theoretically ideal but suffers from the problem that circuits performing extensions of many of the basic 1- and 2-Qbit operations to the 5-Qbit codewords are cumbersome. The current favorite is a 7-Qbit code, devised by Andrew Steane, which permits implementations of many basic operations on codewords, which are not only quite simple but also themselves susceptible to error correction.

The Steane code uses six mutually commuting operators to diagnose the error syndrome:

$$\begin{aligned} \mathbf{M}_0 &= \mathbf{X}_0\mathbf{X}_4\mathbf{X}_5\mathbf{X}_6, & \mathbf{N}_0 &= \mathbf{Z}_0\mathbf{Z}_4\mathbf{Z}_5\mathbf{Z}_6, \\ \mathbf{M}_1 &= \mathbf{X}_1\mathbf{X}_3\mathbf{X}_5\mathbf{X}_6, & \mathbf{N}_1 &= \mathbf{Z}_1\mathbf{Z}_3\mathbf{Z}_5\mathbf{Z}_6, \\ \mathbf{M}_2 &= \mathbf{X}_2\mathbf{X}_3\mathbf{X}_4\mathbf{X}_6, & \mathbf{N}_2 &= \mathbf{Z}_2\mathbf{Z}_3\mathbf{Z}_4\mathbf{Z}_6. \end{aligned} \tag{5.42}$$

The six operators in (5.42) clearly square to give the unit operator. The $\mathbf{M}_i$ trivially commute among themselves as do the $\mathbf{N}_i$, and each $\mathbf{M}_i$ commutes with each $\mathbf{N}_j$, in spite of the anticommutation of each $\mathbf{X}_k$ with the corresponding $\mathbf{Z}_k$, because in every case they share an even number of such pairs. A circuit that measures the six operators (5.42) is shown in Figure 5.9.

The 7-Qbit codewords are defined by

$$\begin{aligned} |\overline{0}\rangle &= 2^{-3/2}(\mathbf{1}+\mathbf{M}_0)(\mathbf{1}+\mathbf{M}_1)(\mathbf{1}+\mathbf{M}_2)|0\rangle_7, \\ |\overline{1}\rangle &= 2^{-3/2}(\mathbf{1}+\mathbf{M}_0)(\mathbf{1}+\mathbf{M}_1)(\mathbf{1}+\mathbf{M}_2)\overline{\mathbf{X}}|0\rangle_7, \end{aligned} \tag{5.43}$$

where

$$\overline{\mathbf{X}} = \mathbf{X}_0\mathbf{X}_1\mathbf{X}_2\mathbf{X}_3\mathbf{X}_4\mathbf{X}_5\mathbf{X}_6, \tag{5.44}$$

so that

$$|1111111\rangle = \overline{\mathbf{X}}|0000000\rangle. \tag{5.45}$$

We again defer our discussion of how to produce these states until after our discussion of how they are used in error correction.

The two states in (5.43) are orthogonal, since each $\mathbf{M}$ flips four Qbits while $\overline{\mathbf{X}}$ flips all seven of them, so the first state is a superposition of 7-Qbit states with an odd number of zeros while the second is a superposition with an even number of zeros. They are normalized to unity, for essentially the same reasons as in the case of 5-Qbit code.

Since $\overline{\mathbf{X}}$ commutes with all the $\mathbf{M}_i$, a general superposition of the two codewords can be written as

$$|\Psi\rangle = \alpha|\overline{0}\rangle + \beta|\overline{1}\rangle = \big(\alpha\mathbf{1} + \beta\overline{\mathbf{X}}\big)|\overline{0}\rangle, \tag{5.46}$$

and its corruption (5.18) assumes the form

$$|e\rangle|\Psi\rangle \to \left(|d\rangle\mathbf{1} + \sum_{i=1}^{7}\big[\,|a_i\rangle\mathbf{X}_i + |b_i\rangle\mathbf{Y}_i + |c_i\rangle\mathbf{Z}_i\big]\right)|\Psi\rangle. \tag{5.47}$$

Because the $\mathbf{M}_i$ all commute and $\mathbf{M}_i(\mathbf{1}+\mathbf{M}_i) = \mathbf{1}+\mathbf{M}_i$, and because the $\mathbf{N}_j$ commute with the the $\mathbf{M}_i$ and with $\overline{\mathbf{X}}$ and have $|0000000\rangle$ as an eigenstate with eigenvalue 1, it follows that $|\overline{0}\rangle$, $|\overline{1}\rangle$, and the general superposition (5.46) are eigenstates of each of the $\mathbf{M}_i$ and $\mathbf{N}_i$ with eigenvalue 1. The 21 possible corruptions of (5.46) appearing in (5.47) are also eigenstates, distinguished by the possible sets of eigenvalues ±1 that the three $\mathbf{M}_i$ and three $\mathbf{N}_i$ can have. As in the 5-Qbit case, this is because each $\mathbf{X}_i$, $\mathbf{Y}_i$, and $\mathbf{Z}_i$ commutes or anticommutes with each of the $\mathbf{M}_i$ and $\mathbf{N}_i$, so each state appearing in (5.47) is indeed an eigenstate of each $\mathbf{M}_i$ and $\mathbf{N}_i$ with eigenvalue 1 or -1.

To see why the results of the six measurements of the $\mathbf{M}_i$ and $\mathbf{N}_i$ determine a unique one of the 22 terms in (5.47), examine Table 5.3, which indicates by a bullet (•) whether an $\mathbf{X}_i$ appears in each of the $\mathbf{M}_i$ and whether a $\mathbf{Z}_i$ appears in each of the $\mathbf{N}_i$. Each $\mathbf{M}_i$ commutes with every $\mathbf{X}_j$; it anticommutes with $\mathbf{Y}_j$ and $\mathbf{Z}_j$ if a bullet appears in the column associated with $\mathbf{X}_j$ and commutes if there is no bullet; each $\mathbf{N}_i$ commutes with every $\mathbf{Z}_j$; it anticommutes with $\mathbf{X}_j$ and $\mathbf{Y}_j$ if a bullet appears in the column associated with $\mathbf{Z}_j$ and commutes if there is no bullet.

Table 5.3. The six error-syndrome operators $\mathbf{M}_i$ and $\mathbf{N}_i$, $i = 0, 1, 2$, for the 7-Qbit code. A bullet (•) indicates whether a given $\mathbf{X}_i$ appears in each $\mathbf{M}_i$ and whether a given $\mathbf{Z}_i$ appears in each $\mathbf{N}_i$.

	$\mathbf{X}_0$	$\mathbf{X}_1$	$\mathbf{X}_2$	$\mathbf{X}_3$	$\mathbf{X}_4$	$\mathbf{X}_5$	$\mathbf{X}_6$
$\mathbf{M}_0$	•				•	•	•
$\mathbf{M}_1$		•		•		•	•
$\mathbf{M}_2$			•	•	•		•
	$\mathbf{Z}_0$	$\mathbf{Z}_1$	$\mathbf{Z}_2$	$\mathbf{Z}_3$	$\mathbf{Z}_4$	$\mathbf{Z}_5$	$\mathbf{Z}_6$
$\mathbf{N}_0$	•				•	•	•
$\mathbf{N}_1$		•		•		•	•
$\mathbf{N}_2$			•	•	•		•

The signature of an $\mathbf{X}_i$ error (or no error) is that all three $\mathbf{M}_i$ measurements give $+1$. The pattern of -1 eigenvalues in the $\mathbf{N}_i$ measurements then determines which of the seven possible $\mathbf{X}_i$ characterize the error. (If all three $\mathbf{N}_i$ measurements also give $+1$ there is no error.)

In the same way, the signature of a $\mathbf{Z}_i$ error (or no error) is that all three $\mathbf{N}_i$ measurements give $+1$ and then the pattern of -1 eigenvalues in the $\mathbf{M}_i$ measurements determines which of the seven possible $\mathbf{Z}_i$ characterize the error.

Finally, the signature of a $\mathbf{Y}_i$ error is that at least some of both the $\mathbf{M}_i$ and the $\mathbf{N}_i$ measurements give -1. The resulting pattern of -1 eigenvalues (which will be the same for both the $\mathbf{M}_i$ and the $\mathbf{N}_i$ measurements) then determines which of the seven possible $\mathbf{Y}_i$ characterize the error.

So the six measurements project the corrupted state into a unique one of the 22 terms in (5.47) and establish which term it is. One can then undo the corruption by applying the appropriate one of the 22 operators $\mathbf{1}$, $\mathbf{X}_0, \ldots, \mathbf{Z}_6$.

To produce the 7-Qbit codewords one cannot immediately extend the method we used in Section 5.5 to produce the 5-Qbit codewords, because the two 7-Qbit codewords and their 21 1-Qbit corruptions constitute only 44 mutually orthogonal states, while the space of seven Qbits has dimension $2^7 = 128$. One can, however, provide the missing 84 dimensions by noting the following.

The $2 \times 7 \times 6 = 84$ states given by

$$\mathbf{X}_i\mathbf{Z}_j|\overline{0}\rangle \quad \text{and} \quad \mathbf{X}_i\mathbf{Z}_j|\overline{1}\rangle, \quad i \neq j, \tag{5.48}$$

are also easily verified to be eigenstates of all the $\mathbf{M}_i$ and $\mathbf{N}_i$. These states can be associated with some of the possible 2-Qbit errors, but this is not pertinent to the use to which we put them here. Like the 1-Qbit $\mathbf{Y}_i$ errors, these states result in at least some of both the $\mathbf{M}_i$ and the $\mathbf{N}_i$ measurements giving -1, but unlike the $\mathbf{Y}_i$ errors, the resulting pattern of -1 eigenvalues will *not* be the same for both the $\mathbf{M}_i$ and the $\mathbf{N}_i$ measurements, since $i \neq j$. Each of the $7 \times 6 = 42$ possibilities for $\mathbf{X}_i\mathbf{Z}_j$ leads to its own characteristic pattern of $+1$ and -1 eigenvalues.

This gets us back to the situation we encountered in the 5-Qbit case. By measuring the seven mutually commuting operators $\mathbf{M}_i$, $\mathbf{N}_i$, and

$$\overline{\mathbf{Z}} = \mathbf{Z}_0\mathbf{Z}_1\mathbf{Z}_2\mathbf{Z}_3\mathbf{Z}_4\mathbf{Z}_5\mathbf{Z}_6, \tag{5.49}$$

we can produce from seven Qbits in an arbitrarily chosen state a unique one of the 128 mutually orthogonal states given by $|\overline{0}\rangle$, $|\overline{1}\rangle$, their 42 different 1-Qbit corruptions, and their 84 different special kinds of 2-Qbit corruptions. The results of the measurement tell us the character (if any) of the corruption, from which we know what operators ($\mathbf{X}_i$, $\mathbf{Y}_i$, $\mathbf{Z}_i$, or $\mathbf{X}_i\mathbf{Z}_j$, possibly combined with $\overline{\mathbf{X}}$) we must apply to the post-measurement state to convert it into $|\overline{0}\rangle$.

A simpler way to produce 7-Qbit codewords is to start with seven Qbits in the standard initial state $|0\rangle_7$, and then measure $\mathbf{M}_0$, $\mathbf{M}_1$, and $\mathbf{M}_2$. The resulting state will be one of the eight states

$$2^{-3/2}(\mathbf{1} \pm \mathbf{M}_0)(\mathbf{1} \pm \mathbf{M}_1)(\mathbf{1} \pm \mathbf{M}_2)|0\rangle_7, \tag{5.50}$$

with the specific pattern of $+$ and $-$ signs being revealed by the measurement. The upper part of Table 5.3 now permits one to choose a unique $\mathbf{Z}_i$ that commutes or anticommutes with each $\mathbf{M}_i$ depending on whether it appears in (5.50) with a $+$ or a $-$ sign. Since $\mathbf{Z}_i|0\rangle_7 = |0\rangle_7$, acting on the seven Qbits with that particular $\mathbf{Z}_i$ converts their state to

$$2^{-3/2}(\mathbf{1} + \mathbf{M}_0)(\mathbf{1} + \mathbf{M}_1)(\mathbf{1} + \mathbf{M}_2)|0\rangle_7 = |\overline{0}\rangle. \tag{5.51}$$

In Section 5.8 we examine a surprisingly simple circuit that encodes a general 1-Qbit state into a 7-Qbit codeword state in the manner of Figure 5.1, without using any measurement gates.

5.7 Operations on 7-Qbit codewords

The virtue of the 7-Qbit code, that makes it preferable to the 5-Qbit code in spite of its greater expenditure of Qbits, is that many of the fundamental 1- and 2-Qbit gates are trivially extended to 7- and 14-Qbit gates acting on the codewords. Because, for example, $\overline{\mathbf{X}}$ commutes with the $\mathbf{M}_i$ and flips all seven Qbits, it implements the logical NOT on the codewords (5.43):

$$\overline{\mathbf{X}}|\overline{0}\rangle = |\overline{1}\rangle, \qquad \overline{\mathbf{X}}|\overline{1}\rangle = |\overline{0}\rangle. \tag{5.52}$$

Similarly, $\overline{\mathbf{Z}}$ commutes with the $\mathbf{M}_i$, anticommutes with $\overline{\mathbf{X}}$, and leaves $|0\rangle_7$ invariant, so it implements the logical $\mathbf{Z}$ on the codewords:

$$\overline{\mathbf{Z}}|\overline{0}\rangle = |\overline{0}\rangle, \qquad \overline{\mathbf{Z}}|\overline{1}\rangle = -|\overline{1}\rangle. \tag{5.53}$$

This much works equally well for the 5-Qbit code. More remarkably, for the 7-Qbit code the bitwise Hadamard transformation,

$$\overline{\mathbf{H}} = \mathbf{H}_0\mathbf{H}_1\mathbf{H}_2\mathbf{H}_3\mathbf{H}_4\mathbf{H}_5\mathbf{H}_6, \tag{5.54}$$

also implements the logical Hadamard transformation on the codewords:

$$\overline{\mathbf{H}}|\overline{0}\rangle = \tfrac{1}{\sqrt{2}}\big(|\overline{0}\rangle + |\overline{1}\rangle\big), \qquad \overline{\mathbf{H}}|\overline{1}\rangle = \tfrac{1}{\sqrt{2}}\big(|\overline{0}\rangle - |\overline{1}\rangle\big). \tag{5.55}$$

(This does not hold for the 5-Qbit code.)

To see this, note first that two normalized states $|\phi\rangle$ and $|\psi\rangle$ are identical if and only if their inner product is 1. (For one can always express $|\psi\rangle$ in the form $|\psi\rangle = \alpha|\phi\rangle + \beta|\chi\rangle$, where $|\chi\rangle$ is orthogonal to $|\phi\rangle$ and $|\alpha|^2 + |\beta|^2 = 1$. We then have $\langle\phi|\psi\rangle = \alpha$, so if $\langle\phi|\psi\rangle = 1$, then $\alpha = 1$ and $\beta = 0$.) Since $|\overline{0}\rangle$ and $|\overline{1}\rangle$ are normalized and orthogonal and since $\overline{\mathbf{H}}$ is unitary and therefore preserves the normalization of $|\overline{0}\rangle$ and $|\overline{1}\rangle$, the four states appearing in the two equalities in (5.55) are all normalized. Therefore, to establish those equalities it suffices to show that

$$1 = \tfrac{1}{\sqrt{2}}\big(\langle\overline{0}|\overline{\mathbf{H}}|\overline{0}\rangle + \langle\overline{0}|\overline{\mathbf{H}}|\overline{1}\rangle\big), \qquad 1 = \tfrac{1}{\sqrt{2}}\big(\langle\overline{1}|\overline{\mathbf{H}}|\overline{0}\rangle - \langle\overline{1}|\overline{\mathbf{H}}|\overline{1}\rangle\big). \tag{5.56}$$

This in turn would follow if we could show that the matrix of the encoded Hadamard in the encoded states is the same as the matrix of the 1-Qbit Hadamard in the 1-Qbit states:

$$\langle\overline{0}|\overline{\mathbf{H}}|\overline{0}\rangle = \langle\overline{0}|\overline{\mathbf{H}}|\overline{1}\rangle = \langle\overline{1}|\overline{\mathbf{H}}|\overline{0}\rangle = \tfrac{1}{\sqrt{2}}, \qquad \langle\overline{1}|\overline{\mathbf{H}}|\overline{1}\rangle = -\tfrac{1}{\sqrt{2}}. \tag{5.57}$$

To establish (5.57), note that it follows from the definition (5.43) of the codewords $|\overline{0}\rangle$ and $|\overline{1}\rangle$ that the four matrix elements appearing in (5.57) are

$$\begin{aligned}\langle\overline{x}|\overline{\mathbf{H}}|\overline{y}\rangle = 2^{-3}\,{}_7\langle 0|\overline{\mathbf{X}}^x(\mathbf{1}+\mathbf{M}_0)(\mathbf{1}+\mathbf{M}_1)(\mathbf{1}+\mathbf{M}_2)\overline{\mathbf{H}}(\mathbf{1}+\mathbf{M}_0)\\ \times(\mathbf{1}+\mathbf{M}_1)(\mathbf{1}+\mathbf{M}_2)\overline{\mathbf{X}}^y|0\rangle_7.\end{aligned} \tag{5.58}$$

Since $\mathbf{HX} = \mathbf{ZH}$ and $\mathbf{XH} = \mathbf{HZ}$, and since each $\mathbf{N}_i$ differs from $\mathbf{M}_i$ only by the replacement of each $\mathbf{X}$ by the corresponding $\mathbf{Z}$, it follows that

$$\overline{\mathbf{H}}\,\mathbf{M}_i = \mathbf{N}_i\,\overline{\mathbf{H}}, \qquad \mathbf{M}_i\,\overline{\mathbf{H}} = \overline{\mathbf{H}}\,\mathbf{N}_i. \tag{5.59}$$

So we can bring all three terms $\mathbf{1}+\mathbf{M}_i$ in (5.58) on the right of $\overline{\mathbf{H}}$ over to the left if we replace each by $\mathbf{1}+\mathbf{N}_i$. But since the $\mathbf{M}$s and $\mathbf{N}$s all commute we can then bring all three terms $\mathbf{1}+\mathbf{M}_i$ on the left of $\overline{\mathbf{H}}$ over to the right if we again replace each by $\mathbf{1}+\mathbf{N}_i$. The effect of these interchanges is simply to change all the $\mathbf{M}$s in (5.58) into $\mathbf{N}$s:

$$\begin{aligned}\langle\overline{x}|\overline{\mathbf{H}}|\overline{y}\rangle = 2^{-3}\,{}_7\langle 0|\overline{\mathbf{X}}^x(\mathbf{1}+\mathbf{N}_0)(\mathbf{1}+\mathbf{N}_1)(\mathbf{1}+\mathbf{N}_2)\overline{\mathbf{H}}(\mathbf{1}+\mathbf{N}_0)\\ \times(\mathbf{1}+\mathbf{N}_1)(\mathbf{1}+\mathbf{N}_2)\overline{\mathbf{X}}^y|0\rangle_7.\end{aligned} \tag{5.60}$$

Since each $\mathbf{N}_i$ commutes with $\overline{\mathbf{X}}$ (there are four anticommutations) we have

$$\langle\overline{x}|\overline{\mathbf{H}}|\overline{y}\rangle = 2^{-3}\,{}_7\langle 0|(\mathbf{1}+\mathbf{N}_0)(\mathbf{1}+\mathbf{N}_1)(\mathbf{1}+\mathbf{N}_2)\overline{\mathbf{X}}^{x}\,\overline{\mathbf{H}}\,\overline{\mathbf{X}}^{y}(\mathbf{1}+\mathbf{N}_0) \\ \times(\mathbf{1}+\mathbf{N}_1)(\mathbf{1}+\mathbf{N}_2)|0\rangle_7, \qquad (5.61)$$

but since each $\mathbf{N}_i$ acts as the identity on $|0\rangle_7$, each of the six $\mathbf{1}+\mathbf{N}_i$ can be replaced by a factor of 2, reducing (5.61) simply to

$$\langle\overline{x}|\overline{\mathbf{H}}|\overline{y}\rangle = 2^{3}\,{}_7\langle 0|\overline{\mathbf{X}}^{x}\,\overline{\mathbf{H}}\,\overline{\mathbf{X}}^{y}|0\rangle_7. \qquad (5.62)$$

Since $\overline{\mathbf{X}}$, $\overline{\mathbf{H}}$, and $|0\rangle_7$ are tensor products of the seven 1-Qbit quantities $\mathbf{X}$, $\mathbf{H}$, and $|0\rangle$, (5.62) is just

$$\langle\overline{x}|\overline{\mathbf{H}}|\overline{y}\rangle = 2^{3}\,\langle x|\mathbf{H}|y\rangle^{7}. \qquad (5.63)$$

But since

$$\langle 0|\mathbf{H}|0\rangle = \langle 0|\mathbf{H}|1\rangle = \langle 1|\mathbf{H}|0\rangle = \tfrac{1}{\sqrt{2}}, \qquad \langle 1|\mathbf{H}|1\rangle = -\tfrac{1}{\sqrt{2}}, \quad (5.64)$$

(5.63) does indeed reduce to (5.57), establishing that $\overline{\mathbf{H}} = \mathbf{H}^{\otimes 7}$ does indeed act as a logical Hadamard gate on the codewords.

Nor is it difficult to make a 14-Qbit logical cNOT gate that takes the pair of codewords $|\overline{x}\rangle|\overline{y}\rangle$ into $|\overline{x}\rangle|\overline{x\oplus y}\rangle$. One simply applies ordinary cNOT gates to each of the seven pairs of corresponding Qbits in the two codewords. This works because each of the codewords in (5.43) is left invariant by each of the $\mathbf{M}_i$. If the control codeword is in the state $|\overline{0}\rangle$ then the pattern of flips applied to the target codeword for each of the eight terms in the expansion of the control codeword

$$|\overline{0}\rangle = 2^{-3/2}\big(\mathbf{1}+\mathbf{M}_0+\mathbf{M}_1+\mathbf{M}_2+\mathbf{M}_1\mathbf{M}_2+\mathbf{M}_2\mathbf{M}_0 \\ +\mathbf{M}_0\mathbf{M}_1+\mathbf{M}_0\mathbf{M}_1\mathbf{M}_2\big)|0\rangle_7 \qquad (5.65)$$

is simply given by the corresponding product of $\mathbf{M}_i$. Since each $\mathbf{M}_i$ acts as the identity on both $|\overline{0}\rangle$ and $|\overline{1}\rangle$, the target codeword is unchanged. On the other hand, if the control codeword is in the state $|\overline{1}\rangle$ then the pattern of flips applied to the target codeword differs from this by an additional application of $\overline{\mathbf{X}}$, which has precisely the effect of interchanging $|\overline{0}\rangle$ and $|\overline{1}\rangle$.

Because of the simplicity of all these encoded gates, one can use error correction to eliminate malfunctions of the elementary gates themselves, if the rate of malfunctioning is so low that only a single one of the seven elementary gates is likely to malfunction. In the case of the 1-Qbit encoded gates, their elementary components act only on single Qbits in the codeword, so if only a single one of them malfunctions then only a single Qbit in the codeword will be corrupted and the error-correction procedure described above will restore the correct output. But this works as well for the encoded cNOT gate, since if only a single one of the elementary 2-Qbit cNOT gates malfunctions,

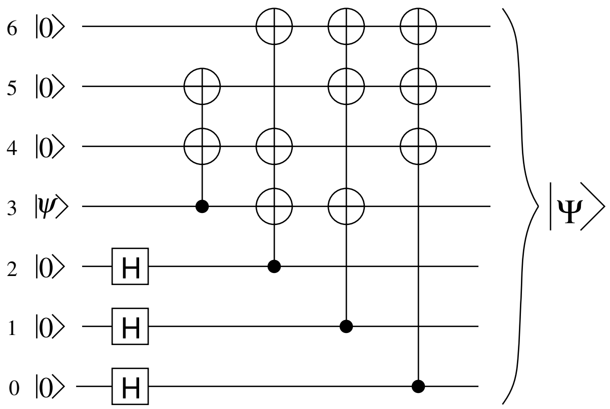


Fig 5.10 A 7-Qbit encoding circuit (a) that takes $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ into the corresponding superposition of the two 7-Qbit codewords given in (5.43), $|\Psi\rangle = \alpha|\overline{0}\rangle + \beta|\overline{1}\rangle$. The numbering of the Qbits from 6 to 0 is made explicit to facilitate comparison with the form (5.42)–(5.44) of the codewords.

this will affect only single Qbits in each of the two encoded 7-Qbit words, and the correct output will again be restored by applying error correction to both of the codewords.

Another virtue of codeword gates that can be constructed as tensor products of uncoded gates is that they cannot (when functioning correctly) convert single-Qbit errors to multiple-Qbit errors, as more elaborate constructions of codeword gates might do. This highly desirable property is called *fault tolerance*. The great advantage of the 7-Qbit code is that many of the most important logical gates can be implemented in a fault-tolerant way.

5.8 A 7-Qbit encoding circuit

The circuit in Figure 5.10 encodes a general 1-Qbit state into a 7-Qbit codeword without using any measurement gates, in a manner analogous to the way Figure 5.1 produces 3-Qbit codewords.

Since the circuit is unitary and therefore linear, it is enough to show that it works when $|\psi\rangle = |0\rangle$ and when $|\psi\rangle = |1\rangle$. This follows from the fact that if the $(n+1)$-Qbit gate $\mathbf{C}^U$ is a controlled n-Qbit unitary $\mathbf{U}$, then

$$\begin{aligned}\mathbf{C}^U\big(\mathbf{H}|0\rangle\big) \otimes |\Phi\rangle_n &= \mathbf{C}^U \tfrac{1}{\sqrt{2}}\big(|0\rangle + |1\rangle\big) \otimes |\Phi\rangle_n \\ &= \tfrac{1}{\sqrt{2}}\big(\mathbf{1} + \mathbf{X} \otimes \mathbf{U}\big)|0\rangle \otimes |\Phi\rangle_n, \qquad (5.66)\end{aligned}$$

where the control Qbit is on the left. If this is applied to the three controlled triple-NOT gates in Figure 5.10 then, reading from left to right, the resulting operations are $(1/\sqrt{2})(\mathbf{1} + \mathbf{X}_2\mathbf{X}_3\mathbf{X}_4\mathbf{X}_6) =$

$(1/\sqrt{2})(\mathbf{1}+\mathsf{M}_2)$, $(1/\sqrt{2})(\mathbf{1}+\mathsf{X}_1\mathsf{X}_3\mathsf{X}_5\mathsf{X}_6) = (1/\sqrt{2})(\mathbf{1}+\mathsf{M}_1)$, and $(1/\sqrt{2})(\mathbf{1}+\mathsf{X}_0\mathsf{X}_4\mathsf{X}_5\mathsf{X}_6) = (1/\sqrt{2})(\mathbf{1}+\mathsf{M}_0)$.

When $|\psi\rangle = |0\rangle$ the controlled double-NOT on the left acts as the identity, so the circuit does indeed produce the codeword $|\overline{0}\rangle$ in (5.43). When $|\psi\rangle = |1\rangle$, the controlled double-NOT on the left acts as $\mathsf{X}_4\mathsf{X}_5$. The circuit after that action is exactly the same as when $|\psi\rangle = |0\rangle$, except that the initial state of Qbits 3, 4, and 5 on the left is $|1\rangle$ rather than $|0\rangle$. Since all X_i commute, the state that results is not $|\overline{0}\rangle$ but $\mathsf{X}_3\mathsf{X}_4\mathsf{X}_5|\overline{0}\rangle$. But

$$\mathsf{X}_3\mathsf{X}_4\mathsf{X}_5 = \mathsf{X}_0\mathsf{X}_1\mathsf{X}_2\mathsf{X}_3\mathsf{X}_4\mathsf{X}_5\mathsf{X}_6\mathsf{M}_0\mathsf{M}_1\mathsf{M}_2 = \overline{\mathsf{X}}\mathsf{M}_0\mathsf{M}_1\mathsf{M}_2. \qquad (5.67)$$

Since $\mathsf{M}_0\mathsf{M}_1\mathsf{M}_2$ acts as the identity on $|\overline{0}\rangle$, the resulting state is indeed $|\overline{1}\rangle = \overline{\mathsf{X}}|\overline{0}\rangle$.

A less direct method to confirm that Figure 5.10 produces the 7-Qbit encoding, analogous to the method described in Section 5.9 for the 5-Qbit encoding, is given in Appendix O.

5.9 A 5-Qbit encoding circuit

The circuit in Figure 5.11 encodes a general 1-Qbit state into a 5-Qbit codeword without using any measurement gates.

The circuit differs from one reported by David DiVincenzo[2] only by the presence of the 1-Qbit gates **ZHZ** on the left. When $|\psi\rangle = |x\rangle$ DiVincenzo's circuit produces two orthogonal linear combinations of the codewords (5.43), which are, of course, equally valid choices. But to get the codewords in (5.43) one needs these additional gates. (I have written them in the symmetric form **ZHZ** rather than in the simpler equivalent form **YH** both to spare the reader from having to remember that $\mathsf{Y} = \mathsf{ZX}$ and not **XZ**, and also to spare her the confusion of having to reverse the order of gates when going from a circuit diagram to the corresponding equation.)

In contrast to the superficially similar circuit for the 7-Qbit code in Figure 5.10, there does not seem to be a transparently simple way to demonstrate that the circuit in Figure 5.11 does produce the 5-Qbit codewords. One can always, of course, write down the action of each successive gate in the circuit, and check that the resulting unwieldy expressions are identical to the explicit expansions of the codewords (5.33) in computational-basis states. A less clumsy proof follows from the fact that $|\overline{0}\rangle$ is the unique (to within an overall phase factor $e^{i\varphi}$) joint eigenvector with all eigenvalues 1 of the five mutually commuting operators consisting of the four error-syndrome operators $\mathsf{M}_0, \ldots, \mathsf{M}_3$

[2] David P. DiVincenzo, "Quantum Gates and Circuits," *Proceedings of the Royal Society of London* A **454**, 261–276 (1998), `http://arxiv.org/abs/quant-ph/9705009`.

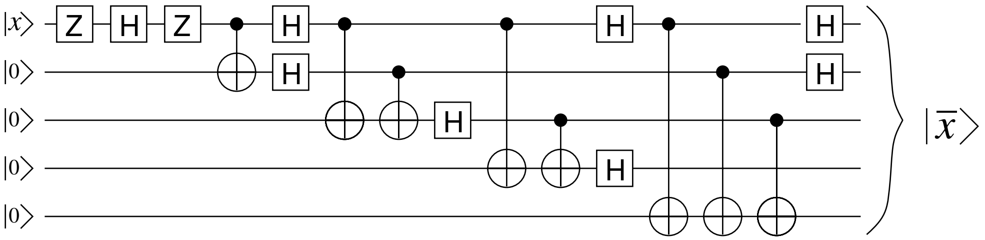


Fig 5.11 A 5-Qbit encoding circuit. If the initial state of the Qbit on the top wire is $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, then the circuit produces the corresponding superposition of the two 5-Qbit codewords given in (5.33), $|\Psi\rangle = \alpha|\overline{0}\rangle + \beta|\overline{1}\rangle$. This fact is established in Figures 5.12–5.20. The figure illustrates this for the states $|0\rangle$ and $|1\rangle$ ($x = 0$ or 1) on the upper wire. Since a product of unitary gates is linear, the circuit encodes arbitrary superpositions of these states.

of Equation (5.29) and the operator $\overline{\mathbf{Z}}$ of Equation (5.38). So if we can establish that the state $|\overline{x}\rangle$ produced in Figure 5.11 is invariant under the four $\mathbf{M}_i$, that it is invariant under $\overline{\mathbf{Z}}$ when $x = 0$, and that applying the 5-Qbit $\overline{\mathbf{X}}$ to $|\overline{x}\rangle$ is the same as applying the 1-Qbit $\mathbf{X}$ to $|x\rangle$, then we will have shown that the circuit produces the 5-Qbit encoding to within an overall phase factor $e^{i\varphi}$. Having done this, we can then confirm that $e^{i\varphi} = 1$ by evaluating the projection on $|0\rangle_5$ of the state produced by the circuit when $x = 0$.

To learn the actions of various products of 1-Qbit $\mathbf{X}$s and $\mathbf{Z}$s on the state produced by the circuit in Figure 5.11, we apply them on the right side of the diagram, and then bring them to the left through the cNOT gates and 1-Qbit gates that make up the circuit, until they act directly on the input state on the left. In doing this we must use the fact that bringing an $\mathbf{X}$ (or a $\mathbf{Z}$) through a Hadamard converts it to a $\mathbf{Z}$ (or an $\mathbf{X}$), bringing an $\mathbf{X}$ through a $\mathbf{Z}$ introduces a factor of -1, and bringing an $\mathbf{X}$ or a $\mathbf{Z}$ through a cNOT has the results shown in Figure 5.12: a $\mathbf{Z}$ on the control Qbit (or an $\mathbf{X}$ on the target Qbit) commutes with a cNOT, while bringing a $\mathbf{Z}$ through the target Qbit (or an $\mathbf{X}$ through the control Qbit) introduces an additional $\mathbf{Z}$ on the control Qbit (or $\mathbf{X}$ on the target Qbit).

Figure 5.13 uses these elementary facts to show that $\mathbf{M}_0 = \mathbf{Z}_1\mathbf{X}_2\mathbf{X}_3\mathbf{Z}_4$ leaves both codewords invariant, by demonstrating that it can be brought to the left through all the gates in the circuit to act on the input state $|x0000\rangle$ as $\mathbf{Z}_2$. Figures 5.14–5.16 show similar things for $\mathbf{M}_1 = \mathbf{Z}_2\mathbf{X}_3\mathbf{X}_4\mathbf{Z}_0$, $\mathbf{M}_2 = \mathbf{Z}_3\mathbf{X}_4\mathbf{X}_0\mathbf{Z}_1$, and $\mathbf{M}_3 = \mathbf{Z}_4\mathbf{X}_0\mathbf{X}_1\mathbf{Z}_2$, which can be brought to the left through all the gates to act on the input state as $\mathbf{Z}_0$, $\mathbf{Z}_3$, and $\mathbf{Z}_1$. Figure 5.17 shows that $\overline{\mathbf{X}} = \mathbf{X}_0\mathbf{X}_1\mathbf{X}_2\mathbf{X}_3\mathbf{X}_4$ can be brought to the left through all the gates of the circuit to act on the input state $|x0000\rangle$ as $\mathbf{X}_4\mathbf{Z}_2\mathbf{Z}_1$, which simply interchanges $x = 0$ and

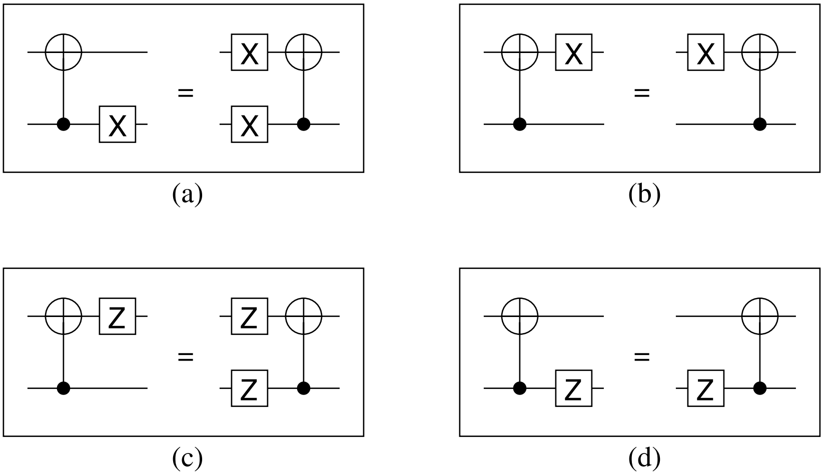


Fig 5.12 Easily verifiable identities useful in determining how various products of **X**s and **Z**s act on the circuit of Figure 5.11. (a) A cNOT can be interchanged with an **X** acting on the control Qbit, if another **X** acting on the target Qbit is introduced. (b) A cNOT commutes with an **X** acting on the target Qbit. (c) A cNOT can be interchanged with a **Z** acting on the target Qbit, if another **Z** acting on the control Qbit is introduced. (d) A cNOT commutes with a **Z** acting on the control Qbit.

$x = 1$, thereby demonstrating that $\overline{\mathbf{X}}$ acts as logical X on the codewords. Figure 5.18 shows the analogous property for $\overline{\mathbf{Z}} = \mathbf{Z}_0\mathbf{Z}_1\mathbf{Z}_2\mathbf{Z}_3\mathbf{Z}_4$, which can be brought to the left through all the gates of the circuit to act on the input state $|x0000\rangle$ as $\mathbf{Z}_4\mathbf{Z}_3\mathbf{Z}_0$, which multiplies it by $(-1)^x$, thereby demonstrating that $\overline{\mathbf{Z}}$ acts as logical Z on the codewords. Finally Figures 5.19 and 5.20 show that the inner product of the codeword state $|\overline{0}\rangle$ with the computational-basis state $|00000\rangle$ is $\frac{1}{4}$, thereby demonstrating that the circuit produces the codewords (5.33) with the right phase.

In Appendix O this circuit-theoretic approach is used to give a second (more complicated, but instructive) demonstration of the validity of the 7-Qbit encoding circuit of Figure 5.10.

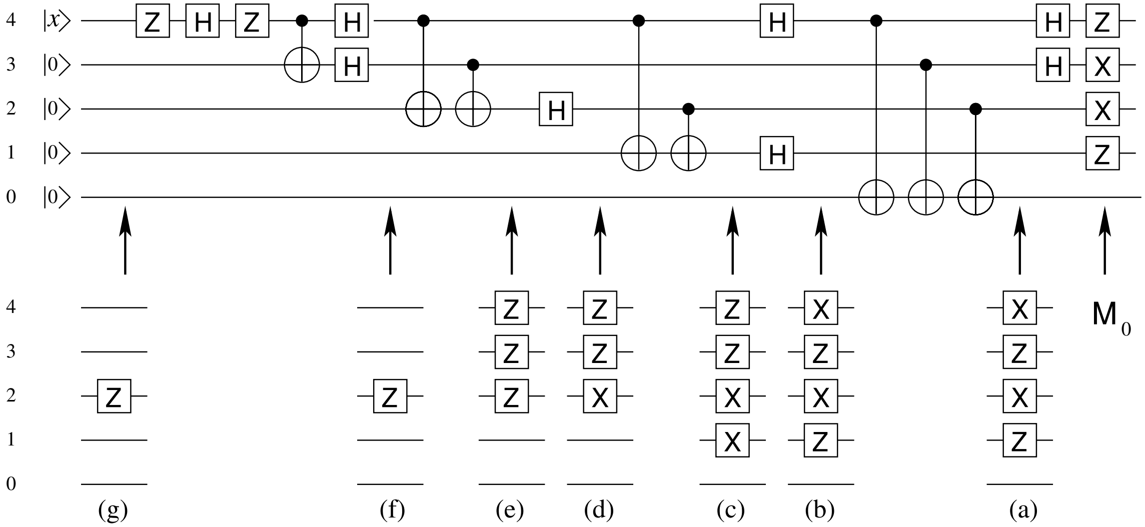


Fig 5.13 Demonstration that $\mathbf{M}_0 = \mathbf{Z}_1\mathbf{X}_2\mathbf{X}_3\mathbf{Z}_4$ acting on the output of the encoding circuit in Figure 5.11 is the same as $\mathbf{Z}_2$ acting on the input, which leaves the input invariant. On the extreme left $\mathbf{M}_0$ is applied to the output of the circuit. The insets (a)–(g) show what happens as the $\mathbf{X}$ and $\mathbf{Z}$ gates making up $\mathbf{M}_0$ are moved to the left through the gates of the circuit. (a) $\mathbf{Z}_4$ and $\mathbf{X}_3$ are changed to $\mathbf{X}_4$ and $\mathbf{Z}_3$ as a result of having been brought through Hadamard gates. (b) Bringing the two $\mathbf{X}$ gates through the control Qbits of cNOT gates produces a pair of cancelling $\mathbf{X}$ gates on the common target Qbit, so the set of gates in (a) is unchanged when it is moved to (b). (c) The Hadamard gates convert $\mathbf{X}_4$ and $\mathbf{Z}_1$ to $\mathbf{Z}_4$ and $\mathbf{X}_1$. (d) Bringing $\mathbf{X}_2$ through the control Qbit of the cNOT produces an $\mathbf{X}$ on its target Qbit which cancels the $\mathbf{X}$ already there. (e) The Hadamard on Qbit 2 converts the $\mathbf{X}$ to a $\mathbf{Z}$. (f) Moving the $\mathbf{Z}_2$ through the targets of the two cNOTs produces $\mathbf{Z}$ gates on their control Qbits which cancel the two $\mathbf{Z}$ gates already there. (g) The resulting $\mathbf{Z}_2$ can be moved all the way to the left.

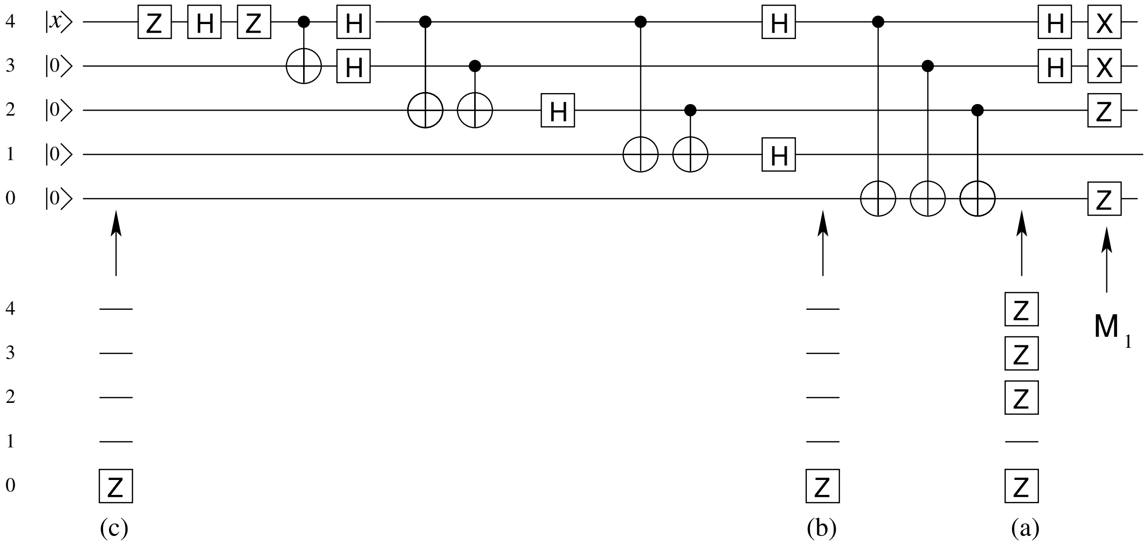


Fig 5.14 Using the identities in Figure 5.12 and the fact that bringing a **Z** through a Hadamard converts it to an **X** and vice versa establishes that $\mathbf{M}_1$ can be brought to the left through the gates of the encoding circuit to act directly on $|x0000\rangle$ as $\mathbf{Z}_0$.

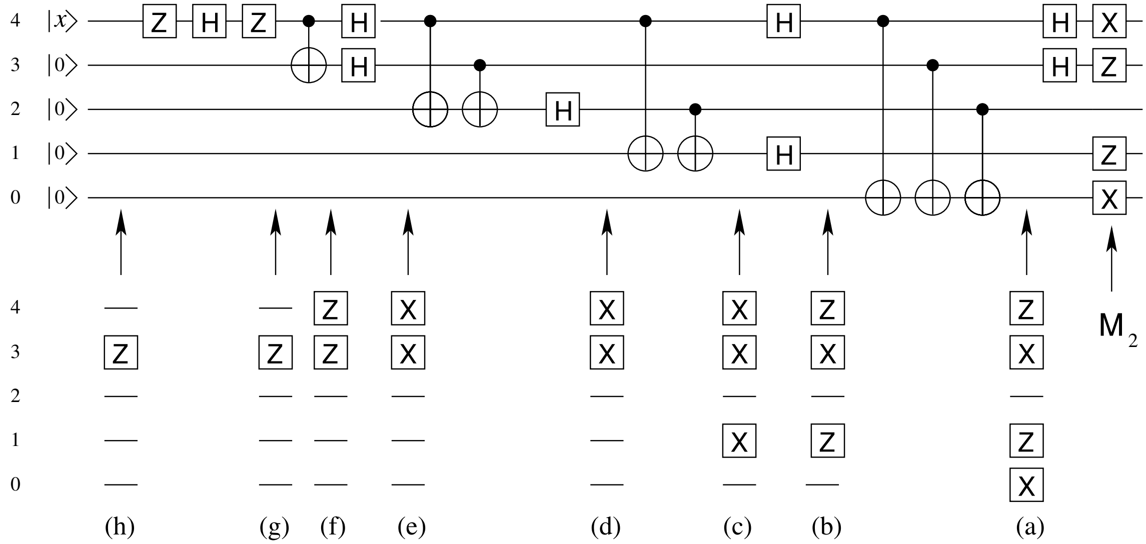


Fig 5.15 $\mathbf{M}_2$ can be brought to the left through the gates of the encoding circuit to act directly on $|x0000\rangle$ as $\mathbf{Z}_3$.

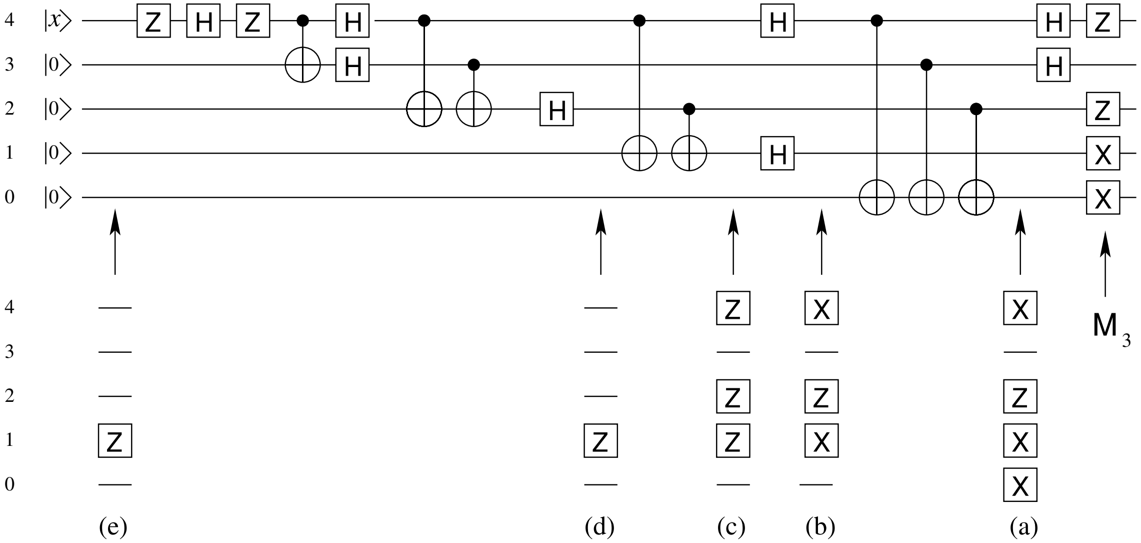


Fig 5.16 $\mathbf{M}_3$ can be brought to the left through the gates of the encoding circuit to act directly on $|x0000\rangle$ as $\mathbf{Z}_1$.

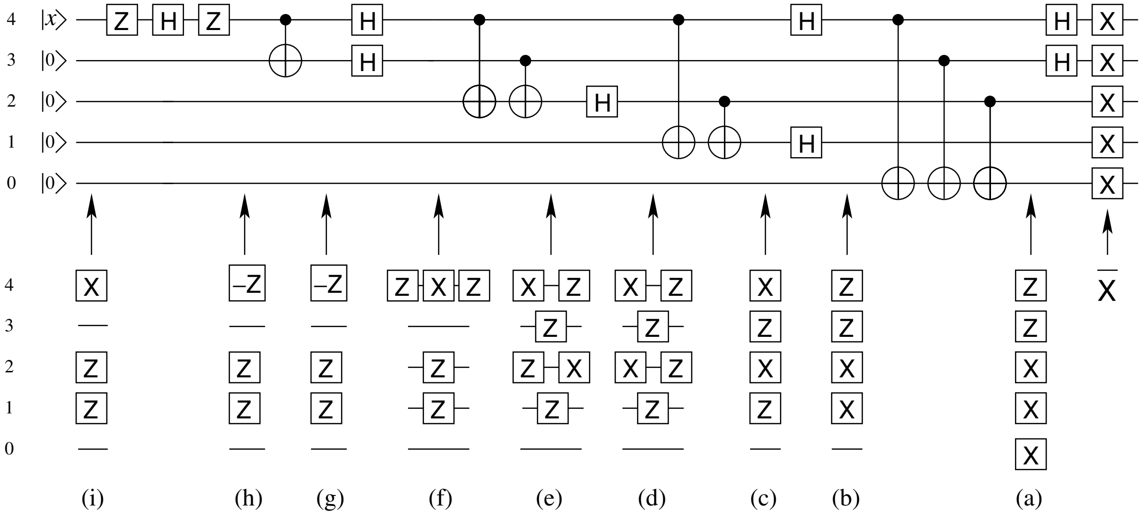


Fig 5.17 Demonstration that $\overline{\mathbf{X}} = \mathbf{X}_0\mathbf{X}_1\mathbf{X}_2\mathbf{X}_3\mathbf{X}_4$ acting on the output of the encoding circuit in Figure 5.11 is the same as $\mathbf{X}_4\mathbf{Z}_2\mathbf{Z}_1$ acting on the input, which interchanges $|00000\rangle$ and $|10000\rangle$. (a) Bringing $\mathbf{X}_4$ and $\mathbf{X}_3$ through the Hadamards converts them to $\mathbf{Z}_4$ and $\mathbf{Z}_3$. (b) Bringing $\mathbf{X}_2$ through the cNOT controlled by Qbit 2 produces an $\mathbf{X}$ on the target Qbit 0, which cancels the $\mathbf{X}$ already there. (c) The Hadamards convert $\mathbf{Z}_4$ and $\mathbf{X}_1$ to $\mathbf{X}_4$ and $\mathbf{Z}_1$. (d) Bringing $\mathbf{X}_4$ and $\mathbf{X}_2$ to the left produces two $\mathbf{X}_1$ gates which cancel; bringing $\mathbf{Z}_1$ to the left then produces additional $\mathbf{Z}_4$ and $\mathbf{Z}_2$ gates. (e) The Hadamard $\mathbf{H}_2$ interchanges the $\mathbf{X}_2$ and $\mathbf{Z}_2$ gates. (f) First bring to the left the $\mathbf{Z}_2$ gate, then the $\mathbf{X}_4$ gate. (g) The $\mathbf{H}_4$ converts $\mathbf{ZXZ}$ to $\mathbf{XZX} = -\mathbf{Z}$. (h) No further changes. (i) $\mathbf{Z}$ commutes with itself, is changed to $\mathbf{X}$ on passing through $\mathbf{H}$, and acquires another minus sign on passage through $\mathbf{Z}$.

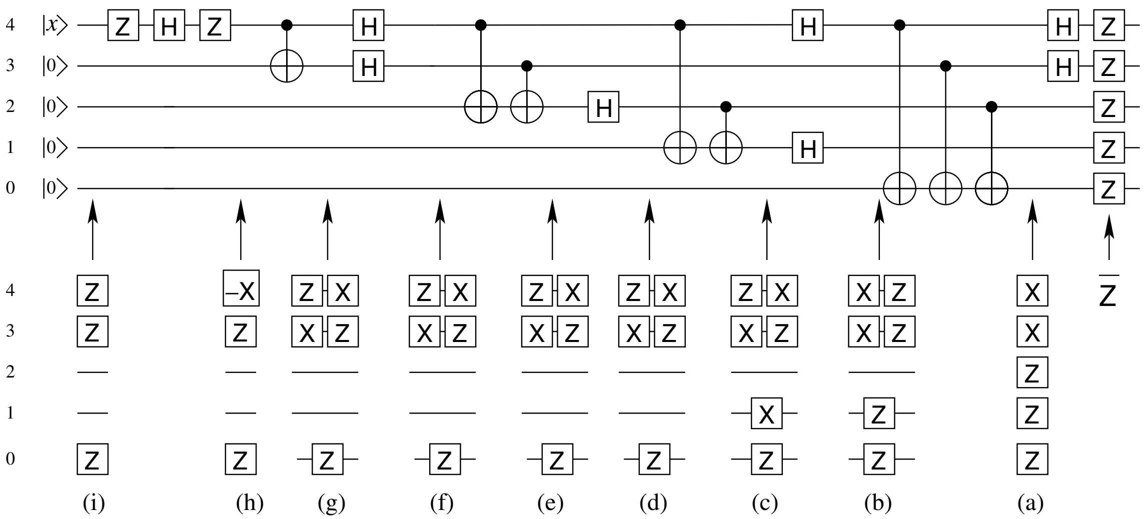


Fig 5.18 Demonstration that $\overline{\mathbf{Z}} = \mathbf{Z}_0\mathbf{Z}_1\mathbf{Z}_2\mathbf{Z}_3\mathbf{Z}_4$ acting on the output of the encoding circuit in Figure 5.11 is the same as $\mathbf{Z}_4\mathbf{Z}_3\mathbf{Z}_0$ acting on the input, which takes $|x0000\rangle$ into $(-1)^x|x0000\rangle$.

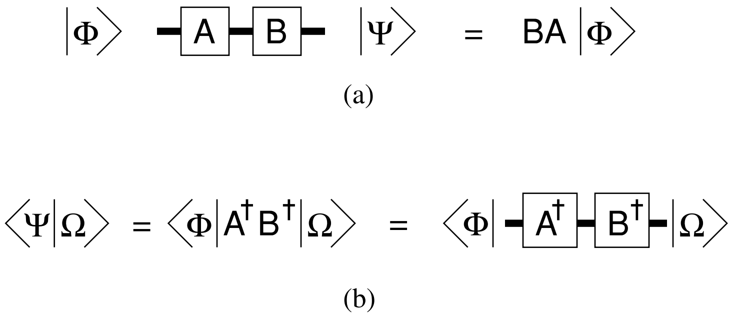


Fig 5.19 A circuit-theoretic way to evaluate inner products. (a) A circuit taking the input $|\Phi\rangle$ into the output $|\Psi\rangle = \mathbf{BA}|\Phi\rangle$. The inner product $\langle\Psi|\Omega\rangle$ of the output state Ψ with some other state $|\Omega\rangle$ is given by $\langle\Phi|\mathbf{A}^\dagger\mathbf{B}^\dagger|\Omega\rangle$. The diagram on the right in (b) shows this inner product being evaluated by first letting $\mathbf{B}^\dagger$ act on $|\Omega\rangle$, then letting $\mathbf{A}^\dagger$ act on the result, and then taking the inner product with the input state $|\Phi\rangle$. Evidently this generalizes to the product of many gates. If the gates are all Hermitian, as they are in the circuit of Figure 5.11, then the circuit on the right of (b) is identical to the circuit on the left of (a). The resulting evaluation of the inner product of $|\Omega\rangle = |0\rangle_5$ with the state $|\overline{0}\rangle$ produced by letting the circuit of Figure 5.11 act on $|\Phi\rangle = |0\rangle_5$ is carried out in Figure 5.20.

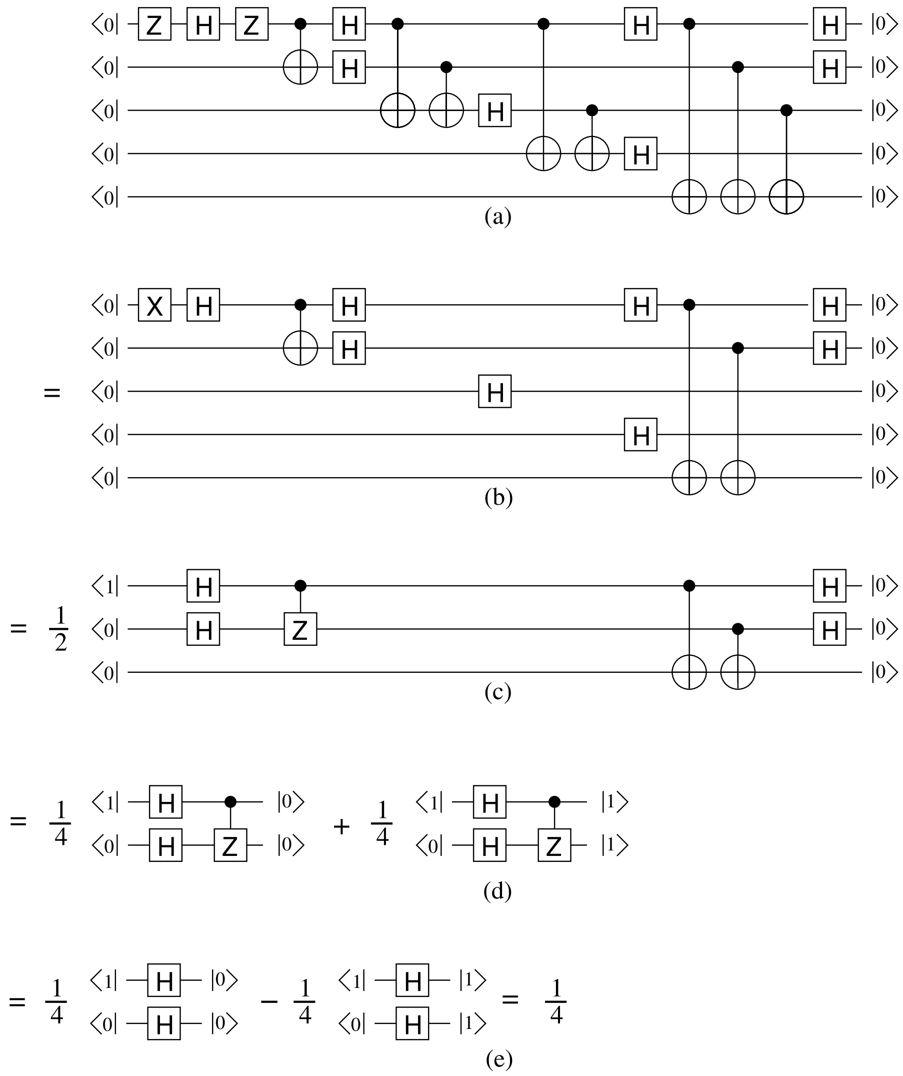


Fig 5.20 Demonstration that the state produced by the encoding circuit in Figure 5.11 when $x = 0$ has an inner product with the state $|0\rangle_5$ that is $\frac{1}{4}$, thereby establishing that the phase factor $e^{i\varphi} = 1$ – i.e. that the state is precisely $|\overline{0}\rangle$ without any additional phase factor. (a) Circuit-theoretic representation of the inner product, following the procedure developed in Figure 5.19; all gates now act to the right. (b) Elimination of operations in (a) that act as the identity: the cNOT on the extreme right of (a) can be dropped since its control Qbit is in the state $|0\rangle$; since $\mathbf{H}|0\rangle$ is invariant under $\mathbf{X}$, the pair of cNOT gates targeting Qbit 1 can be dropped, as can the pair targeting Qbit 2. (c) A pair of Hadamards on Qbit 4 in (b) cancel; a Hadamard on Qbit 3 in (b) is moved to the left converting a cNOT to a controlled-Z; Qbits 2 and 1 in (b) simply give the matrix element $\langle 0|\mathbf{H}|0\rangle = \frac{1}{\sqrt{2}}$, resulting in an overall factor of $\frac{1}{2}$. (d) Expanding both states $\mathbf{H}|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ on the right of (c), the effect of the two cNOT gates in (c) is that only the terms in $|0\rangle|0\rangle$ and $|1\rangle|1\rangle$ give nonzero contributions. (e) The action of the controlled-Z gates in (d) has been carried out, leaving a sum of products of matrix elements of $\mathbf{H}$.