In this paper we show how the fault–tolerant error correction scheme recently proposed by DiVincenzo and Shor may be improved. Our scheme, unlike the earlier one, can also deal with a single error that might occur during the gate operations that are required for the implementation of the error correction and not only in-between the gates and hence presents an improvement towards enabling error correction and with it the practical possibility of some more involved quantum computations, such as e.g. factorization of large numbers.

Soon after the idea of quantum computation became an active part of current research through the seminal work of Shor on factorization [1], decoherence and especially spontaneous emission was recognized as a major problem that cannot be ignored, at least when one is interested in practical applications including especially the factorization of large numbers [2,3]. It has become clear that efficient error correction methods have to be found to overcome decoherence if we are ever to realize the possibility of building a quantum computer. In fact, inspired by the theory of classical error correction [4], Shor, Calderbank and Shor and independently Steane [5–8] proposed the first quantum error correction codes able to correct errors that occur during the storage of qubits. More error correction codes have been discovered and theoretical work undertaken has elucidated the structure of quantum codes [9–24] even further.

However, these investigations have dealt with the problem of processing the stored information and not with the correction of errors that might occur during gate operation. A significant step forward in this direction was made by Shor and DiVincenzo [18,20]. In [18] the idea of fault–tolerant implementation of quantum gates was developed where one error would not lead to many errors, and hence can be corrected by subsequent error correction. An example of a fault–tolerant network for error correction proposed in [20] is shown in fig. 1. It has the important property of being able to perform error correction even if an error occurs between the execution of two of its quantum gates. To be more precise, if the incoming state is error free, and one error occurs during the error correction, then the outgoing state will have at most one error. This is a substantial improvement compared to previous error correcting networks, where one error during error correction usually results in more than one error in the outgoing state. However, the network given in [20] has this fault–tolerant property only for errors that occur between its successive quantum gate operations. Errors during gate operation that are needed to implement the error correcting network will still produce many errors, as an error in a two–bit gate usually leads to two–bit errors.

It is, fortunately, possible to improve this network such that these errors are also dealt with in a fault–tolerant way. This is an important improvement because we are more likely to experience errors during gate operation than in-between, as the interval between successive gates can be made very small compared to the gate operation time.

Before we present our improved protocol for fault–tolerant error correction, we briefly review the procedure given in [20] and show that it fails if errors occur during gate operation. The network under consideration (see fig. 1) performs the error correction for a five–bit, single error correcting code [10,11] with the code–words

\[ |\tilde{0}\rangle = |C_0\rangle + |C_1\rangle \]
\[ |\tilde{1}\rangle = |C_0\rangle - |C_1\rangle \]

(1)

(2)

where

\[ |C_0\rangle = |00000\rangle + |11000\rangle + |01100\rangle + |00110\rangle \]
\[ + |00011\rangle + |10001\rangle - |10100\rangle - |01010\rangle \]
\[ - |00101\rangle - |10010\rangle - |01001\rangle - |11110\rangle \]
\[ - |01111\rangle - |10111\rangle - |11011\rangle - |11101\rangle \]

(3)

and \( |C_1\rangle \) being the state where each qubit is inverted with respect to \( |C_0\rangle \). The fault–tolerant error correcting network for this code is presented in fig. 1 [20]. The incoming encoded state is represented by the top five lines. The lower four lines represent the error syndrome and are prepared in a known state. It should be noted that, to ensure fault–tolerant operation, each line (qubit) of the error syndrome actually consists of four separate qubits, initially prepared in a state with zero parity of the form

\[ |\Psi\rangle = \sum_{n,n':T=0} |n\rangle , \]

(4)

where \( T = (1111) \) and \( n \cdot T \) is the bitwise product modulo 2 [18]. That means that \( |\psi\rangle \) is the equally weighted superposition of all four-qubit states of even parity. The action of the four CNOT operations on a qubit of the syndrome then has to be rearranged as indicated in fig. 2. This ensures that, after an error in one of the CNOT’s,
with the initial syndrome state.

One might think that this task could also be achieved
by performing a measurement on these 16 qubits after
the error correction step itself. Due to the additional
information introduced by this conditional repetition we
are now able to treat errors that occurred during gate
operations. This is possible because although an error
that occurs during a quantum gate introduces a two-bit
error, one of the errors is in the syndrome. The error in
the syndrome, however, does not propagate as each

Table 1: All possible single bit errors and their corre-
sponding syndromes that may occur in the network of
fig. 1 are listed. $X_i$ denotes an amplitude error in bit $i$,
$Y_i$ a phase error in bit $i$ and $Z_i = X_i Y_i$. All syndromes
are different so that it is possible to correct a general single
bit error.

in the ‘corrected’ state in the scheme presented in fig.
1. Assume that an error during the CNOT-operation
between bit 0 and $a_3$ has an effect as if there was an
amplitude error in both bit 0 and bit $a_3$ (in general the
effect will be a superposition of many possible two-bit
errors). Then according to table 1 the error syndrome
would indicate an amplitude error $X_3$ which would sub-
sequently be ‘corrected’ and the outgoing state then has
two amplitude errors in bits 0 and 3! A state with two
errors, however, cannot be dealt with by subsequent er-
ror correction steps which would, in actual fact, add even
more errors to the state. Therefore the error correction
procedure in fig. 1 cannot be regarded as fault-tolerant
if errors occur during the gate operation. This is an im-
portant shortcoming because most errors will occur dur-
ing the quantum gate operation and not in between, as
the time delay between successive quantum gates can be
made much smaller than the time required to perform a
quantum gate.

In the following we will show that the error correction
scheme of DiVincenzo and Shor [20] which we have dis-
cussed above can be improved in order to make sure that
also errors during quantum gate operation can be dealt
with fault-tolerantly. To achieve this we have to repeat
the generation of the above error syndrome conditional
on the result of the first syndrome before we decide on
the error correction step itself. Due to the additional
information introduced by this conditional repetition we
are now able to treat errors that occurred during gate
operations. This is possible because although an error
that occurs during a quantum gate introduces a two-bit
error, one of the errors is in the syndrome. The error in
the syndrome, however, does not propagate as each

---

Fig. 1: Fault–tolerant network given in [20]: $R$ describes
a one bit rotation which takes $|0\rangle \rightarrow (|0\rangle + |1\rangle)/\sqrt{2}$ and
$|1\rangle \rightarrow (|0\rangle - |1\rangle)/\sqrt{2}$. An encircled cross denotes a NOT
operation, while a small circle denotes a control bit. The
first five qubits are encoded in the superposition of $|C_0\rangle$
and $|C_1\rangle$ given in eq. (3), while each of the last four
‘lines’ represents four qubits initially in the state $|\Psi\rangle$ as
in eq. (4). At the end the error syndrome is obtained
by performing a measurement on these 16 qubits after
which appropriate correction is applied to the first five
qubits.

no back-action of errors takes place which would other-
wise lead to multiple errors in the outgoing state [18].
One might think that this task could also be achieved
with the initial syndrome state $|\Psi\rangle = |0000\rangle$ instead of
the one given in eq. (4). This is, however, not so, as
then different code–words would lead to different states
of $|\Psi\rangle$, which would then enable us to single out one su-
perposition state of the code from a measurement of the
state resulting from $|\Psi\rangle$. The state eq. (4), in contrast,
contains only information about its parity. It can now
be checked that the network presented in fig. 1 is fault–
tolerant if errors occur between operations of its quantum
gates [20]. In table 1 we present the possible syndromes
and the related errors, $X_i$ (amplitude error on the $i$–th
bit), $Y_i$ (phase error on the $i$–th bit), and $Z_i = X_i Y_i$.

We now show by means of an example that one error
during the operation of a CNOT can lead to two errors

---

Fig. 2: This diagram represents how the controlled NOTs
from fig.1. are to be applied in order to avoid back–action
of errors.
Table 2: The possible results for the syndrome $S_1$ after at most one error at an arbitrary position in the network in fig. 3 are shown together with the appropriate action that has to be taken for each of the outcomes.

<table>
<thead>
<tr>
<th>Syndrome $S_1$</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_1 = 0$</td>
<td>No correction</td>
</tr>
<tr>
<td>$S_1 \neq 0$</td>
<td>Generate another syndrome and correct error indicated by it</td>
</tr>
</tbody>
</table>

syndrome is measured before the next one is produced. In table 2 the two possible outcomes for the first syndrome are shown together with the appropriate action that has to be taken. Using table 2 it is now simple to check that the network in fig. 1 together with the conditional generation of the error syndrome is capable of fault-tolerant error correction even if an error occurs at an arbitrary position in the error correcting network. It should be noted that we only produce another syndrome if there was an error either in the incoming state or during the error correction network. This, however, means that an error in the construction of the additional syndrome would be a second order effect which we neglect here as we only deal with single error correcting codes. It should also be noted that it is important that the second syndrome is produced conditional on the first one. If we would generate two syndromes from the outset then we could obtain ambiguous results exactly in the case where the first syndrome does not indicate an error but the second one does. This would then require the generation of yet a third syndrome conditional on the outcome of the first two syndromes.

We can summarize the result of our error correction protocol with conditional generation of error syndromes by:

- **If the incoming state is error-free and only one error occurs at an arbitrary position during the network operation, the outgoing state has at most one error. If the incoming state has one error and no further error occurs during the error correction then the outgoing state will be corrected perfectly.**

This allows fault-tolerant error correction if at most one error occurs in the incoming state and the error correction step together. This is achieved using the idea of conditional construction of error syndromes in the ‘fault tolerant’ error correction network given by [20]. The scheme of [20] alone, as we saw, could not cope with the errors occurring during the gate operation. As these errors can be expected to be predominant this is an important shortcoming of the procedure. The error correction protocol with conditional generation of the error syndrome, as presented here, however, is fault–tolerant even if an error occurs during the gate operation, and hence can fault–tolerantly correct for a general single error at arbitrary position. The result that conditional error syndromes can be used to treat errors during gate operation is also interesting because it shows that it is not always necessary to implement the quantum gates in a fault-tolerant way. This can simplify the construction of the error correction networks because although for special, practically important, errors there exist efficient fault-tolerant implementations of quantum gates [25] in general the fault-tolerant implementation of quantum gates can be complicated.

**Acknowledgements**

One of us (MBP) would like to thank D.P. DiVincenzo for useful discussions. This work was supported by the European Community, the UK Engineering and Physical Sciences Research Council and by a Feodor-Lynen grant of the Alexander von Humboldt foundation.

---