7.3 Surface code

The surface code is a QEC code that is related to topology. The idea is that a logical qubit is encoded in $L \times L$ physical qubits as in the layout presented in Fig. 7.12. The array they construct has to be considered with periodic boundary conditions. The L^2 qubits are divided in two classes:

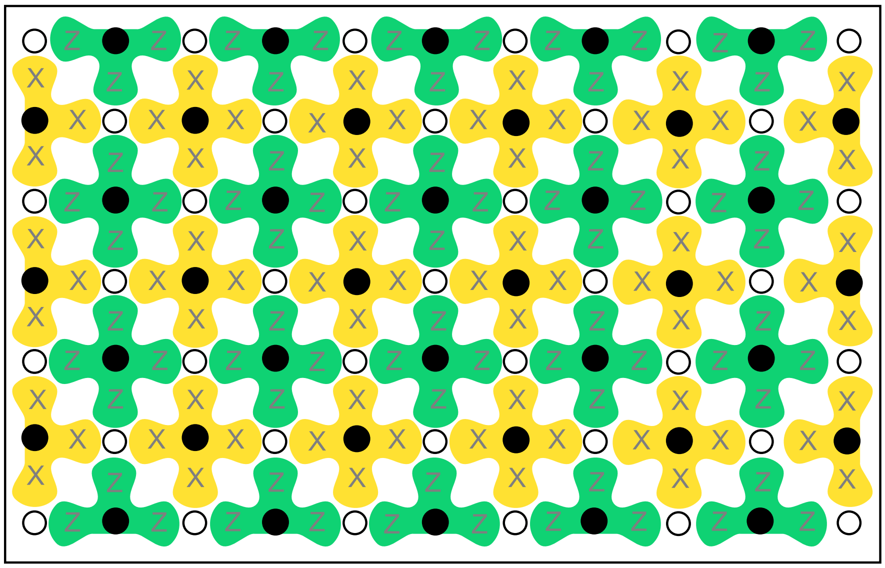
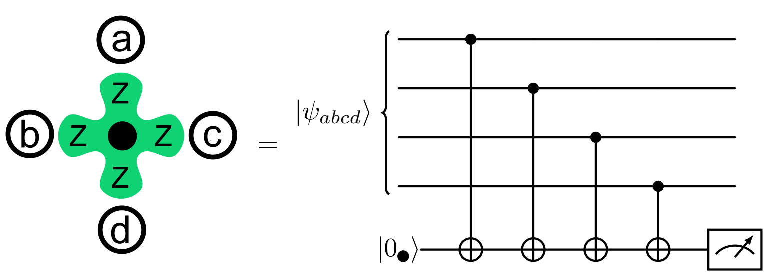


Fig. 7.12: Graphical representation of the L^2 physical qubits (full and open circles) in the array generating the surface code through their interaction (green and yellow connections).

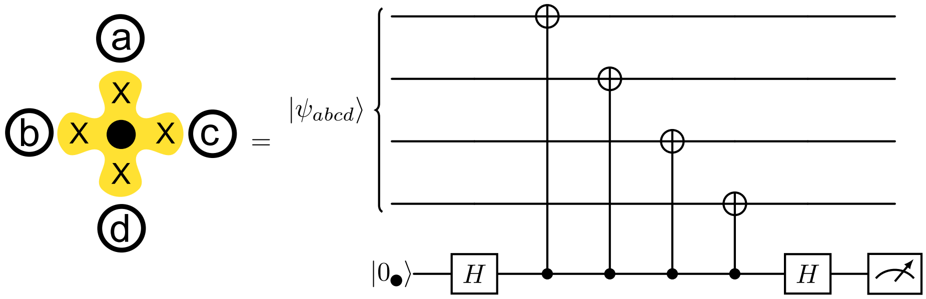
- Half of the physical qubits are used as data qubits: they store quantum states $|\psi_\mathrm{L}\rangle$ that will be used for computation. They are represented with open circles ○.
- Half of the physical qubits are called measurement qubits and they are employed as error detecting qubits. They are represented with full circles ●. There are two types of measurement qubits:
  - Measure Z or Z-syndrome qubits, which are represented in green,
  - Measure X or X-syndrome qubits, which are represented in yellow.

Each data qubit is coupled to two X-syndrome and two Z-syndrome qubits. Each measurement qubit is coupled to four data qubits. These couplings are describe as the following. For the green block, i.e. the Z-syndrome, we have



(7.118)

where $|0_\bullet\rangle$ indicates that the qubit ● has been initiallised in $|0\rangle$. Similarly, for the yellow block one has



(7.119)

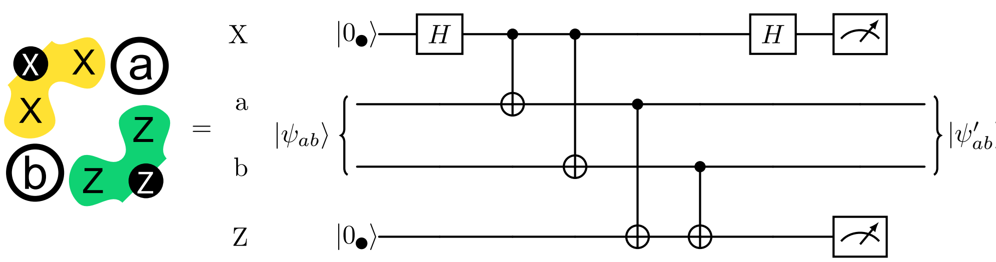
In such a way, the measurement qubits are coupled to the data qubits. These circuits are run in cycles, between one logical operation and the following one, so to keep track of the errors that occur in between.

To understand the logic of the surface code, let us focus on the case where one has only four physical qubits: 2 are data qubits and 2 are measurement qubits (one in X and one in Z). The generators of the stabilisers are the operators $\hat{X}_a\hat{X}_b$ and $\hat{Z}_a\hat{Z}_b$. Here, we employ the notation $\hat{Z}_i$ to identify the $\hat{\sigma}_z^{(i)}$ Pauli operator acting on the i-th physical qubit. One can easily show that these operators commute, i.e. $\left[\hat{X}_a\hat{X}_b, \hat{Z}_a\hat{Z}_b\right] = 0$, although they do not at the level of single qubit, i.e. $\left[\hat{X}_i, \hat{Z}_i\right] \neq 0$. Thus, they have common eigeinstates, which identify the division of the Hilbert space $\mathbb{H}'$ in the code and error subspaces (for the sake of simplicity, we will drop all the normalisation constants):

$\lvert\psi\rangle$	$\hat{X}_a\hat{X}_b$	$\hat{Z}_a\hat{Z}_b$
$\lvert 00\rangle + \lvert 11\rangle$	$+1$	$+1$
$\lvert 00\rangle - \lvert 11\rangle$	-1	$+1$
$\lvert 01\rangle + \lvert 10\rangle$	$+1$	-1
$\lvert 01\rangle - \lvert 10\rangle$	-1	-1

(7.120)

The circuit that applies these two stabilisers is



(7.121)

where the first qubit is the X-syndrome and the last is the Z-syndrome. Considering the generic state $|\psi_{ab}\rangle$ for the qubits a and b being

$$|\psi_{ab}\rangle = A\,|00\rangle + B\,|01\rangle + C\,|10\rangle + D\,|11\rangle\,, \tag{7.122}$$

one can input this state in the circuit in Eq. (7.121) and, before the measurements, obtains that the total state reads

$$\begin{aligned} |\Psi_{XabZ}\rangle &= (A+D)\,|0\rangle\,(|00\rangle + |11\rangle)\,|0\rangle \\ &+ (A-D)\,|1\rangle\,(|00\rangle - |11\rangle)\,|0\rangle \\ &+ (B+C)\,|0\rangle\,(|01\rangle + |10\rangle)\,|1\rangle \\ &+ (B-C)\,|1\rangle\,(|01\rangle - |10\rangle)\,|1\rangle\,. \end{aligned} \tag{7.123}$$

It follows that, after the measurement of the X and Z-syndrome qubits, one obtains — depending on the outcomes $\{\,M_X, M_Z\,\}$ — the following states with the corresponding probabilities $P_{|\psi'_{ab}\rangle}$:

$$\begin{array}{c|c|c} \{\, M_X, M_Z \,\} & |\psi'_{ab}\rangle & P_{|\psi'_{ab}\rangle} \\ \hline \{\, +1, +1 \,\} & |00\rangle + |11\rangle & |A+D|^2 \\ \{\, -1, +1 \,\} & |00\rangle - |11\rangle & |A-D|^2 \\ \{\, +1, -1 \,\} & |01\rangle + |10\rangle & |B+C|^2 \\ \{\, -1, -1 \,\} & |01\rangle - |10\rangle & |B-C|^2 \end{array} \quad (7.124)$$

After the collapse on one of these common eigenstates of $\hat{X}_a\hat{X}_b$ and $\hat{Z}_a\hat{Z}_b$, subsequent applications of the circuit in Eq. (7.121) will provide always — in the assumption of no noise — the same state.

Example 7.4
Consider the state in Eq. (7.123) and suppose the first cycle (which acts effectively as an encoding) provides the measurements $\{\, M_X = -1, M_Z = -1 \,\}$ *and* $|\psi'_{ab}\rangle = |01\rangle - |10\rangle$*. Now,* $|\psi'_{ab}\rangle$ *is equal to* $|\psi_{ab}\rangle$ *in Eq. (7.122) when setting* $A = D = 0$ *and* $B = -C = 1$*. The corresponding output state at the end of the second cycle before the measurement will be* $|\Psi_{XabZ}\rangle = |1\rangle\,(|01\rangle - |10\rangle)\,|1\rangle$*. This has two important implications: 1) the state* $|\psi'_{ab}\rangle$ *remains untouched by the circuit, which is the implementation of the stabilisers:* $\hat{S}_2\hat{S}_1\,|\psi'_{ab}\rangle = |\psi'_{ab}\rangle$*; 2) also the output of the measurement* $\{\, M_X = -1, M_Z = -1 \,\}$ *remains the same.*

This example shows that, without measuring directly $|\psi'_{ab}\rangle$, one can use the output of the measurements to infer the state of the data qubits. Turning the argument upside-down, an error occuring in the data qubits will be identified by the change in the outcomes of the measurements.

We notice that, in the case of 2 data qubits and 2 measurement qubits, one has

- 4 degrees of freedom where to encode a logical state: there are 2 physical qubits having 2 dimensions. The total Hilbert space $\mathbb{H}'$ has 2×2 dimensions.
- 4 constrains from the stabilisers: we have 2 stabilisers and each divides $\mathbb{H}'$ in 2 subspaces.

Then, there are no free degrees of freedom where one can perform logical operations. In order to do that, one needs to impose the length of the array of physical qubits L being an odd number > 1. In such a way, one has 2 free degrees of freedom that can be employed. The minimal array is that having $L = 3$ with 5 data qubits and 4 measurement qubits. Such an array in shown in Fig. 7.13. The circuit corresponding to the stabilisers

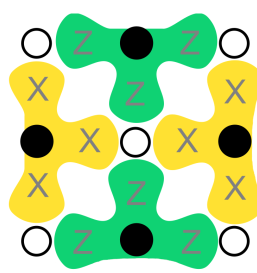
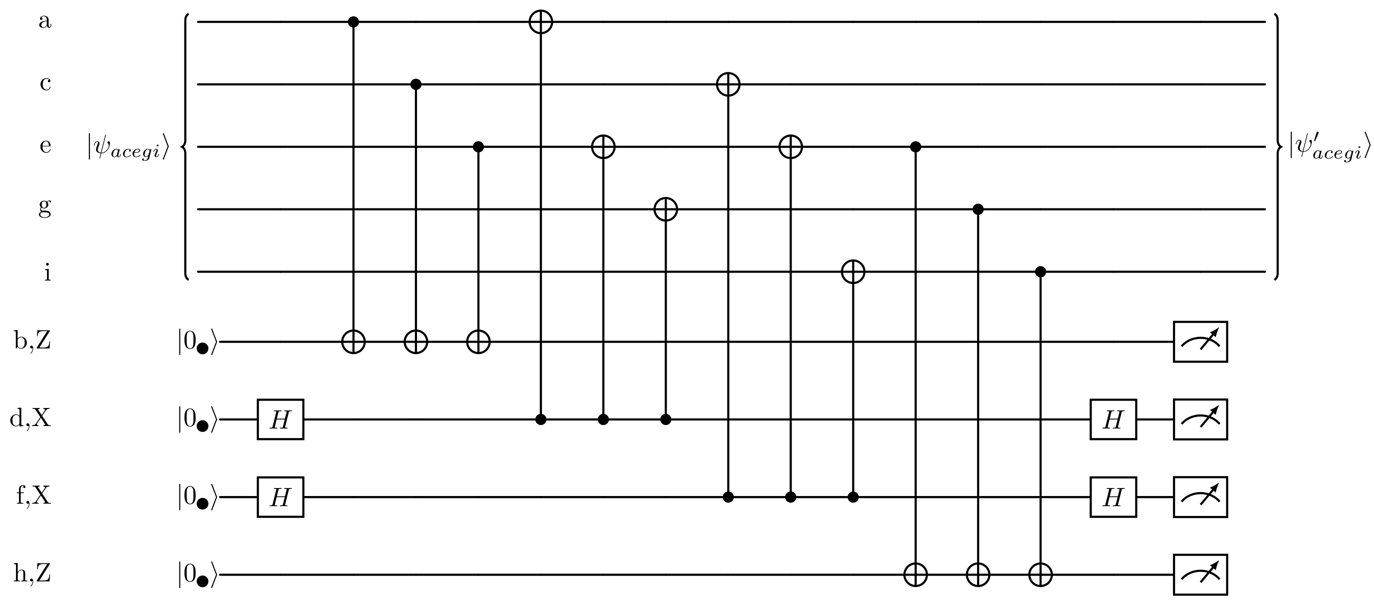


Fig. 7.13: Graphical representation of the surface code with 5 data qubits and 4 measurement qubits.

application is given by



(7.125)

Suppose we input the state $|\psi_{acegi}\rangle = |00000\rangle$. Then, the action of this circuit before the measurement is given by

$$|00000, 0000\rangle \xrightarrow{\text{circuit in Eq. (7.125)}} \begin{aligned} &(|00000\rangle + |10110\rangle + |01101\rangle + |11011\rangle)\,|0000\rangle \\ +&(|00000\rangle - |10110\rangle + |01101\rangle - |11011\rangle)\,|0100\rangle \\ +&(|00000\rangle - |10110\rangle - |01101\rangle + |11011\rangle)\,|0010\rangle \\ +&(|00000\rangle + |10110\rangle - |01101\rangle + |11011\rangle)\,|0110\rangle\,, \end{aligned} \tag{7.126}$$

where we expressed the states in the form $|\psi_{acegi}, \phi_{bdfh}\rangle$. Suppose the measurement results in $\{\, M_b, M_d, M_f, M_h \,\} = \{\, +1, +1, -1, +1 \,\}$, which corresponds to the measurement state $|\phi_{bdfh}\rangle = |0010\rangle$. Then, the data state is given by

$$|\text{data}\rangle = |\psi'_{acegi}\rangle = |00000\rangle - |10110\rangle - |01101\rangle + |11011\rangle\,, \tag{7.127}$$

which remains untouched by subsequent applications of the circuit in Eq. (7.125). This is an easy but lengthy computation if performed in terms of states. However, it becomes trivial and immediate if considering that the circuit corresponds to the application of the stabilisers $\hat{S}_4\hat{S}_3\hat{S}_2\hat{S}_1$ on the state $|\text{data}\rangle$, which has been already stabilised by the same circuit. Thus, $\hat{S}_4\hat{S}_3\hat{S}_2\hat{S}_1\,|\text{data}\rangle = |\text{data}\rangle$.

7.3.1 Detecting errors

There are several kinds of errors that can be detected with surface code. For the sake of simplicity, we consider the case of the array with 5 data and 4 measurement qubits, and that the logical state is encoded in $|\text{data}\rangle$ shown in Eq. (7.127). The latter corresponds to the measurement state $|0010\rangle$, i.e. to the measurement outcomes $\{\, M_b, M_d, M_f, M_h \,\} = \{\, +1, +1, -1, +1 \,\}$. We construct the table of outcomes with respect to the number of cycles that are performed. In the case of no errors and no logical operations, such table reads

# cycles	M_b	M_d	M_f	M_h
1	$+1$	$+1$	-1	$+1$
2	$+1$	$+1$	-1	$+1$
3	$+1$	$+1$	-1	$+1$
4	$+1$	$+1$	-1	$+1$
5	$+1$	$+1$	-1	$+1$
⋮	⋮	⋮	⋮	⋮

(7.128)

We now introduce an error, that will appear exactly at the third cycle. There are also other relevant errors, but we only focuses on the following two kinds.

1) Errors on the measurement or on the syndrome qubits;
These are the errors due to the erroneous output of a measurement M_i, or errors that are applied to the syndrome qubit. The latter will appear as the former. Suppose we have an error on the measurement of the f qubit. Then, the above table becomes

# cycles	M_b	M_d	M_f	M_h
1	$+1$	$+1$	-1	$+1$
2	$+1$	$+1$	-1	$+1$
3	$+1$	$+1$	$+1$	$+1$
4	$+1$	$+1$	-1	$+1$
5	$+1$	$+1$	-1	$+1$
⋮	⋮	⋮	⋮	⋮

(7.129)

where we highlighted the difference between the two tables. The error is only momentaneous. Later applications of the cycle will erase the action of these kind of errors. Indeed, if the error is due to the random erroneous measurement outcome, then the following cycle will (most probably) provide the exact outcome. This will not be the case if there is a systematic error in the measurement process, which cannot be corrected. Conversely, if the error is caused by the application of an external action on the measurement qubit (say the surrounding environment acts with $\hat{Z}$ on the f qubit), then the error vanishes in the next cycle since the qubit's state is initialised at the beginning of each cycle.

2) Errors on the data qubits. These are the errors on the data qubits that can be, for example, due to the surrounding environment, and that can corrupt the information encoded in the data state. It becomes then fundamental to being able to detect and account for such errors for the sake of computation.
Suppose we have a phase-flip error on the e qubit. The state is then transformed as

$$|\psi'_{acegi}\rangle \xrightarrow{\text{phase-flip error on qubit } e} \hat{Z}_e \, |\psi'_{acegi}\rangle \, . \tag{7.130}$$

Then, the Z-syndrome qubits will be unable to detect it. However, X-syndrome qubits d and f will detect the error: their coupling to the data qubit e imposes the action of $\hat{X}_e$, which does not commute with $\hat{Z}_e$. Then, what happens is that the state of the X-syndrome qubits will flip. By supposing that the error occurs at the third cycle. Then, the table of outcomes becomes

# cycles	M_b	M_d	M_f	M_h
1	$+1$	$+1$	-1	$+1$
2	$+1$	$+1$	-1	$+1$
3	$+1$	-1	$+1$	$+1$
4	$+1$	-1	$+1$	$+1$
5	$+1$	-1	$+1$	$+1$
⋮	⋮	⋮	⋮	⋮

(7.131)

Importantly, the outcomes M_d and M_f change sign for any subsequent cycle (if no other errors or logical operations take place). This is how one can distinguish an error on the measurement and on the data qubits. The best way to account for this error is to employ a classical control software that will changes the sign of every subsequent measurement of that data qubit's two adjacent X-syndrome qubits.

When one has an array of larger dimensions, then the situation is more complicated. For example, one might have that several data errors that form paths on the array. If this happens, the errors will be highlighted only by two syndrome qubits at the ends of the error path. An example is shown in Fig. 7.14, where the X-syndrome qubits a and f indicate that an error occurred. However, there is no indication about which path between a and f the Z-errors are covering.

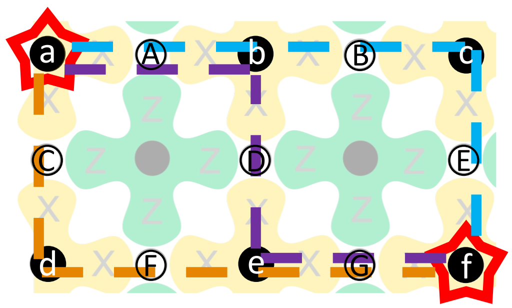


Fig. 7.14: Graphical representation of Z noises detected by the measurement qubits a and f. Different paths (blue, purple and orange) can produce this error syndrome.

This could be one among ABE (blue path), ADG (purpkle path) or CFG (orange path). When accounting via classical control software for the errors, one needs to select one of these paths. The question is what happens if one selects the wrong path? The beauty of the surface code kicks in: as long as the error path and the selected one form a closed loop, the error is well accounted. This is shown with the following argument. Suppose the error path is ABE, which is produced by $\hat{Z}_A\hat{Z}_B\hat{Z}_E$, and we select the path CFG to be corrected, whose correction is given by $\hat{Z}_C\hat{Z}_F\hat{Z}_G$. This is however not a problem, indeed we have that

$$(\hat{Z}_A\hat{Z}_B\hat{Z}_E) = (\hat{Z}_C\hat{Z}_F\hat{Z}_G)\hat{S}_e\hat{S}_d, \tag{7.132}$$

where we defined the stabilisers

$$\hat{S}_e = \hat{Z}_B\hat{Z}_E\hat{Z}_D\hat{Z}_G, \quad \text{and} \quad \hat{S}_d = \hat{Z}_A\hat{Z}_C\hat{Z}_D\hat{Z}_F. \tag{7.133}$$

Therefore, the two errors are related by two stabilisers. This means, that recovery operator $\hat{R}_k$ that corrects $\hat{Z}_A\hat{Z}_B\hat{Z}_E$ can correct also for $\hat{Z}_C\hat{Z}_F\hat{Z}_G$. This has been discussed in Sec. 7.2.5. Thus, every time we can form closed loops, the error can be accounted properly. These are harmless errors.

Conversely, consider now the case shown in Fig. 7.15. Here the error path (shown in orange) crosses the boundary of the array, and due to the periodic boundary conditions only two syndrome qubits highlight the error path. In such a case, one would still be tempted to connect directly the syndrome qubits with a path fully in the array (shown in purple). However, such a correction is not the proper one. Indeed, the two paths would form a logical operation. To visualise harmless and harmful paths, one maps the array on a torus. If the path can be closed, then it is harmless. If the path cannot be closed, then it is harmful. Figure 7.16 shows the mapping between the array and the torus, and highlights the harmless and harmful paths.

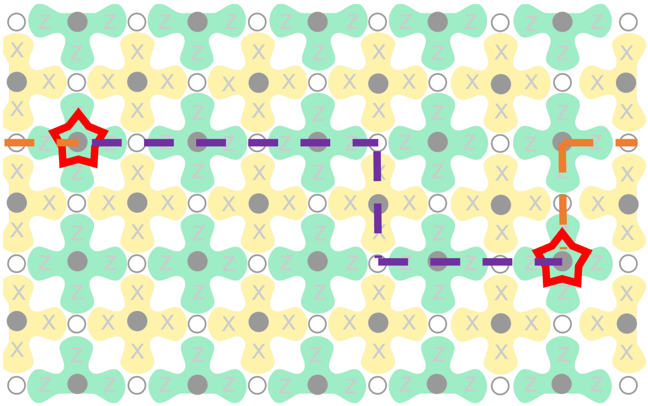


Fig. 7.15: Graphical representation of an error path that crosses the boundary of the array (orange path). If corrected with the purple path, it leads to a logical operation, thus not correcting for the error.

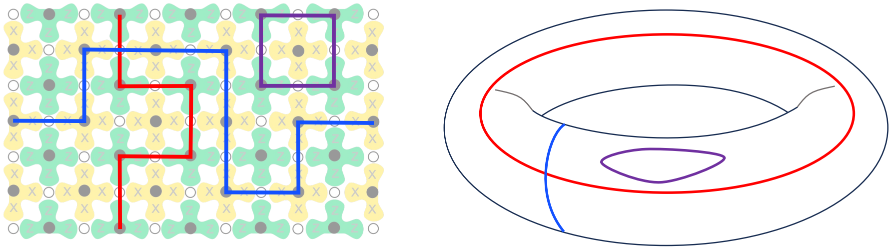


Fig. 7.16: Graphical representation of the mapping of the array on a torus surface. The red path corresponds to a logical X, while the blue path to a logical Z operation. These are harmful error. The purple path is an error that can be corrected.