Quantum Noise and Quantum Error Correction: Description of phase flip error, explanations of the 3-bit and 5- 7- bit stabilizer error correcting codes

Introduction to Quantum Computers Course

ECE TUC

ATHANASIOS KARAKOS

TABLE OF CONTENTS

Introduction

In the real world there are no perfectly *closed* systems, except perhaps the Universe as a whole. Real systems suffer from unwanted interactions with the outside world. These unwanted interactions show up as **noise** in quantum information processing systems. Many *open* quantum and classical systems in widespread use do suffer from a substantial noise problem and in order to encounter with them they make good use of **error correcting techniques.** The key idea is that, if we have a message we wish to protect, we should encode the message by adding some redundant information to the message itself. The simplest encoding method that can be implemented with the help of *repetition codes*. This type of code repeats the message we want to send by a number of times. For example, let's try to send a single bit, given that the probability the bit flips is *p (*1-p the bit remains intact).

3-bit repetition code

Despite the fact that the channel might be noisy, we can ensure the protection by repeating the bit three times. Thus, after the encoding the changes are: **0 -> 000, 1 ->111.** These bit strings are also referred to a ***logical 0*** **and** ***logical 1*****.** After the encoding process we send all 3 bits through the channel. At the receiver's end the output is also three bits and the receiver decodes the current output using the type of decoding, called *majority voting*. The receiver will decide what the original bit was according to how many times *0 or 1* appear most in the output. The receiver fails to decode successfully the message when two or more of the bits sent through the channel were flipped. The probability two or more bits are flipped is expressed as the probability all the bits are flipped or two bits were flipped and one remained intact.

$p_{error} = 3p^2(1-p) - p^3 = 3p^2 - 2p^3$, '3' before p^2 takes all three case where 2 out of 3 bits are flipped. Without encoding the probability of error was p, $\text{if } p < \frac{1}{2} \text{ then } p_{error} < p$

It is shown that the code makes the transmission more reliable!

Quantum Error Correction

Quantum noise causes disruptions to quantum states such as, amplitude damping, bit flip and phase flip errors. It is indisputable that *quantum*

error correcting codes are significant in order to protect a quantum state. To make such codes possible we have to overcome certain obstacles that occur from the singularity of quantum systems.

- ***No cloning***: Because of the no cloning theorem it is forbidden to duplicate the quantum state so we cannot implement the repetition code “quantum mechanically”.
- ***Measurement***: Quantum information is destroyed when the state is measured and makes the recovery impossible.
- ***Continuous errors***: Multiple of different errors may occur on a single qubit and determining the type of error and correcting it may require a vast amount of resources.

Fortunately, none of these problems are unsolvable and experts have designed encoding circuits and stabilizers that detect and correct errors that may happen. The error models we will deal with are consider to act on a single qubit.

3-qubit bit flip code

One of the most fundamental error correcting process is the 3-qubit bit flip code that corrects the bit flip error. Suppose that we have the arbitrary state $|\psi\rangle = a|0\rangle + b|1\rangle$ and with probability *p* the state is taken to $|\psi\rangle \longrightarrow X|\psi\rangle$ (the pauli σx operator, bit flip operator). The encoding circuit that prepares the qubit for the *bit flip channel*, entangles the qubit with auxiliary qubits initialized in ground state 0, using CNOT gates.
Once the initial state $|\psi\rangle = a|0\rangle + b|1\rangle$ has perfectly encoded to $|\psi'\rangle = a|000\rangle + b|111\rangle$

Each of the three qubits is passed through the transmission channel and we perform the next process which is *syndrome diagnosis*. There are four projection operators corresponding to four error syndromes.

$$
\begin{aligned}
P_0 &\equiv |000\rangle\langle 000| + |111\rangle\langle 111| : \text{no error} \\
P_1 &\equiv |100\rangle\langle 100| + |011\rangle\langle 011| : \text{bit flip on 1st qubit} \\
P_2 &\equiv |010\rangle\langle 010| + |101\rangle\langle 101| : \text{bit flip on 2nd qubit} \\
P_3 &\equiv |001\rangle\langle 001| + |110\rangle\langle 110| : \text{bit flip on 3rd qubit}
\end{aligned}
\qquad \textbf{(1)}
$$

We perform a measurement that tells what error occurred on the encoded state. Suppose we have the corrupted output state $|\psi'\rangle = a|100\rangle + b|011\rangle$

Each operator acts and the probability a bit flipped, results from $\langle\psi'|P_j|\psi'\rangle$

If the result is 1, that means the j-qubit is flipped, if it's 0 it didn't flip. For j = 0,2,3 $\langle\psi'|P_j|\psi'\rangle$ = 0. For j =1 :

$$\langle\psi'|P_1|\psi'\rangle = (a\langle 100| + b\langle 011|)(a|100\rangle + b|011\rangle) = |a|^2 + |b|^2 = 1$$

The system detected error on the first qubit. Now we can proceed to the recovery part. Similarly to detecting there are four correcting operators that act according to the previous measurement result.

$$S_0 = I \otimes I \otimes I : \text{do nothing}$$
$$S_1 = X \otimes I \otimes I : \text{flip 1st qubit}$$
$$S_2 = I \otimes X \otimes I : \text{flip 2nd qubit}$$
$$S_3 = I \otimes I \otimes X : \text{flip 3rd qubit} \quad \textbf{(2)}$$

This error correction procedure works perfectly for bit flips that occur only on one or fewer qubits. The probability one or fewer bit flips is equivalent to 1 – the probability 2 or 3 qubits flip.

$$p_{error} \equiv p_{\text{no error}} + p_{\text{1 error}} \equiv 1 - p_{\text{2,3 errors}}$$
$$p_{error} = (1-p)^3 + 3p(1-p)^2 = 1 - (3p^2 - 2p^3)$$

Once again $\text{if } p < \frac{1}{2} \text{ then } p_{error} < p$

Let's observe a circuit that implements the 3-qubit bit flip code.

Suppose that we have the initial state $|\psi\rangle = \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$

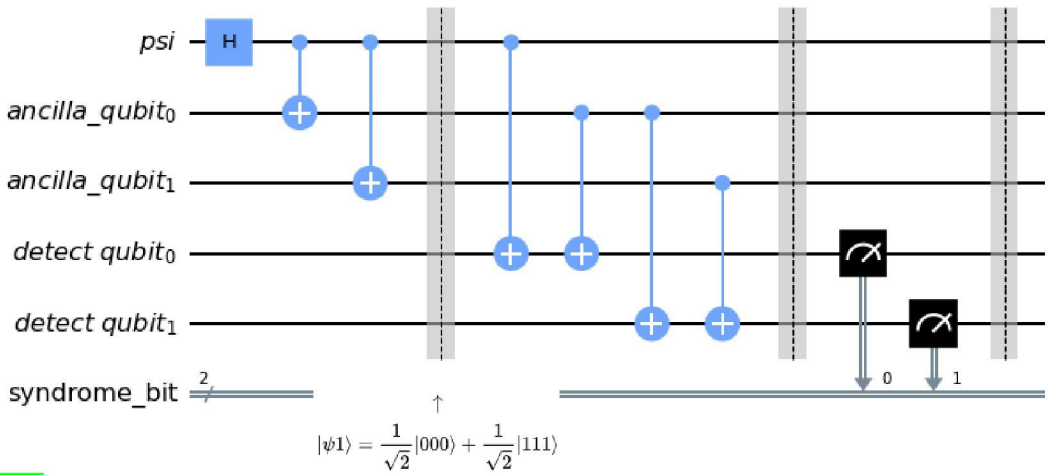


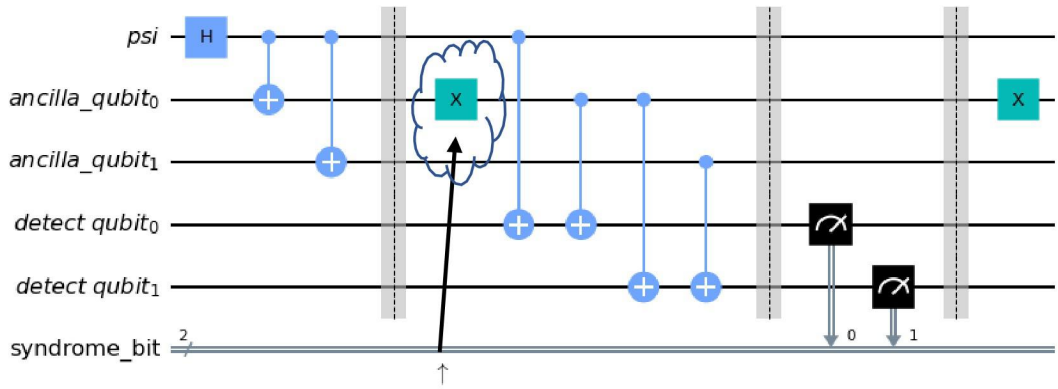
FIG.1. In a real quantum circuit the *P* operators are a process, where we have 2 auxiliary qubits and each one of them is entagled with 2 out of 3 qubits with CNOT gates. The first detect-qubit compares the first two ancillary_qubits, the second compares the second and the third ancillary_qubit. Then we measure the two detect qubits and according

to the measurement result we decide which one of the ***S*** operators to apply. Each measurement result tells us if those 2 qubits we compare are different**. If the result is 0 then those 2 qubits are the same, if it is 1 then they are different. If syndrome_bit_0 = 1 that means there is an error either in the 1st or the 2nd qubit and if syndrome_bit_1 = 0 that means the 2nd and the 3rd qubits are the same so the we identify the 1st qubit as the wrong one**.

Measurement -> *Sj* :
$00 \longrightarrow S_0$ $10 \longrightarrow S_1$ $11 \longrightarrow S_2$ $01 \longrightarrow S_3$

(3)

For example, if the second qubit flips (the bit flip operator *X* is applied)



$$|\psi'\rangle = \frac{1}{\sqrt{2}}|010\rangle + \frac{1}{\sqrt{2}}|101\rangle$$

FIG.2.

The result after the measurement will be:

$$\textbf{syndrome\_bit}_0 = \textbf{syndrome\_bit}_1 = 1$$

for the first detect qubit:

Control is the first qubit of $|\psi'\rangle$ and target is the detect qubit

$$CNOT(|\psi'\rangle\,|detect\_qubit_0\rangle) = \frac{1}{\sqrt{2}}|010\rangle\,|0\rangle + \frac{1}{\sqrt{2}}|101\rangle\,|1\rangle$$

Now the second qubit of $|\psi'\rangle$ controls the CNOT

$$CNOT(|\psi'\rangle\,|detect\_qubit_0\rangle) = \frac{1}{\sqrt{2}}|010\rangle\,|1\rangle + \frac{1}{\sqrt{2}}|101\rangle\,|1\rangle$$

$$|detect\_qubit_0\rangle = |1\rangle$$

for the 2nd detect qubit:

Control is the second qubit of $|\psi'\rangle$ and target is the detect qubit

$$CNOT(|\psi'\rangle\,|detect\_qubit_1\rangle) = \frac{1}{\sqrt{2}}|010\rangle\,|1\rangle\,|1\rangle + \frac{1}{\sqrt{2}}|101\rangle\,|1\rangle\,|0\rangle$$

Now the third qubit of $|\psi'\rangle$ controls the CNOT

$$CNOT(|\psi'\rangle\,|detect\_qubit_1\rangle) = \frac{1}{\sqrt{2}}|010\rangle\,|1\rangle\,|1\rangle + \frac{1}{\sqrt{2}}|101\rangle\,|1\rangle\,|1\rangle$$

In conclusion as we can see (FIG.2.), a *X* gate is applied on the 2nd qubit (ancilla_qubit_0) in order to correct the bit flip error, so the operator *S2* applied **for the measurement result 11.**

Phase flip errors & 3-qubit phase flip code

Apart from bit flip errors, quantum noise could disturb a qubit's phase. It is very interesting how we achieve fault tolerance against a *phase flip error* for a single qubit. In this error model the probability the phase of the qubits flips is *p* and with probability *1-p* is left untouched. Namely the phase flip operator *Z* is applied with probability *p* and the state is taken $a|0\rangle + b|1\rangle \longrightarrow a|0\rangle - b|1\rangle$ under the influence of the phase flip channel.

Moreover, when we work in the basis $|+\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}}$, $|-\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}}$

The operator *Z* takes $|+\rangle$ to $|-\rangle$ and vice versa. It acts like a bit flip operator, but instead of 0 and 1 we have + and - ! Taking into consideration the simple change between the +,- labels the *Z* gate acts, it is useful to encode the arbitrary state of $|\psi\rangle$ using the states $|0_L\rangle \equiv |+++\rangle$, $|1_L\rangle \equiv |---\rangle$ for protection against the phase flip errors. At this point all the steps : **encoding, syndrome detection, error correction**, are performed just as the **bit flip channel**, with respect to the basis $|+\rangle$ & $|-\rangle$. ***Let's demonstrate the 3-qubit phase flip code***. The first step of the encoding is the same as the bit flip the $|\psi\rangle = a|0\rangle + b|1\rangle$ have the arbitrary quantum state

and it is entagled with 2 ancillary qubits, using CNOT gates. The decoding for the phase flip correction has an extra step to accomplish the basis change. *Hadamard (H)* gates are applied to the 3 qubits and the inverse of Hadamard, also the *H* gate, is applied at another point of the procedure, to ensure the change back to the basis $|0\rangle \text{ and } |1\rangle$.

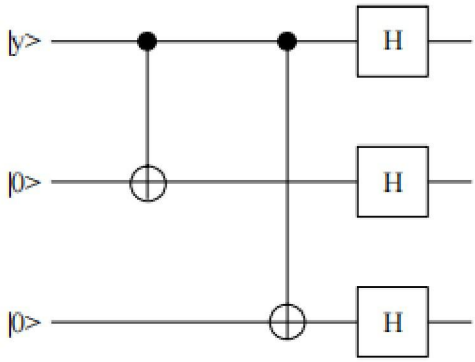


FIG .3. Encoding circuit for the 3-qubit phase flip code.

$$|\psi\rangle = a\,|0\rangle + b\,|1\rangle \longrightarrow |\psi\rangle = a\,|000\rangle + b\,|111\rangle \longrightarrow |\psi\rangle = a\,|+++\rangle + b\,|---\rangle$$

Once the encoding completes we perform the same error detection procedure as before (1) but the projection operators are slightly different, they are conjugated by *Hadamard* gates.

$$P_0' = H^{\otimes 3} P_0 H^{\otimes 3} : \text{No error}$$
$$P_1' = H^{\otimes 3} P_1 H^{\otimes 3} : \textbf{phase flip error on the 1st qubit}$$
$$P_2' = H^{\otimes 3} P_2 H^{\otimes 3} : \textbf{phase flip error on the 2nd qubit}$$
$$P_3' = H^{\otimes 3} P_3 H^{\otimes 3} : \textbf{phase flip error on the 3rd qubit} \quad \textbf{(4)}$$

The measurement result of $\langle\psi|P_j'|\psi\rangle$ will reveal the qubit that has the wrong phase. If the result is **1** for a value of j, then the j-qubit has flipped phase, else it isn't interrupted by the noisy channel. The recovery is accomplished by applying the *Z* gate (or *HXH)* at the qubit the errored happened.

$$S_0 : \textbf{do nothing}$$
$$S_1 = Z \otimes I \otimes I = HXH \otimes I \otimes I : \textbf{phase flip on 1st qubit}$$
$$S_2 = I \otimes Z \otimes I = I \otimes HXH \otimes I : \textbf{phase flip on 2nd qubit}$$
$$S_3 = I \otimes I \otimes Z = I \otimes I \otimes HXH : \textbf{phase flip on 3rd qubit} \quad \textbf{(5)}$$

Now we see how the algorithm works!

Let's take the encoded qubit $|\psi\rangle = a\,|+++\rangle + b\,|---\rangle$ and suppose that an error occurred and the *Z* gate applied to the 2nd qubit, $|\psi'\rangle = a\,|+-+\rangle + b\,|-+-\rangle$

All the measurement results are 0 except one.

$$|\psi\rangle = a|0\rangle + b|1\rangle \longrightarrow |\psi\rangle = a|000\rangle + b|111\rangle \longrightarrow |\psi'\rangle = a|+-+\rangle + b|-+-\rangle$$
$$P_2^{'}|\psi'\rangle = H^{\otimes 3}P_2H^{\otimes 3}(a|+-+\rangle + b|-+-\rangle) = H^{\otimes 3}P_2(a|010\rangle + b|101\rangle) =$$
$$= H^{\otimes 3}(|010\rangle\langle 010| + |101\rangle\langle 101|)(a|010\rangle + b|101\rangle) =$$
$$= H^{\otimes 3}(a|010\rangle + b|101\rangle) = a|+-+\rangle + b|-+-\rangle$$

$$M = \langle\psi'|P_2^{'}|\psi'\rangle = (a\langle +-+| + b\langle -+-|)(a|+-+\rangle + b|-+-\rangle) =$$
$$= |a|^2 + |b|^2 = 1$$

Error on the 2nd qubit detected just as we assumed at the beginning. We will know correct it we the ideal operator *S*.

$$S_2|\psi'\rangle = I \otimes Z \otimes I|\psi'\rangle = a|+\rangle Z|-\rangle|+\rangle + b|-\rangle Z|+\rangle|-\rangle =$$
$$= a|+++\rangle + b|---\rangle$$

Circuit representation

Let's see how a phase flip code implements the steps in a real quantum circuit to detect the error in the previous example, with a = b = $\frac{1}{\sqrt{2}}$

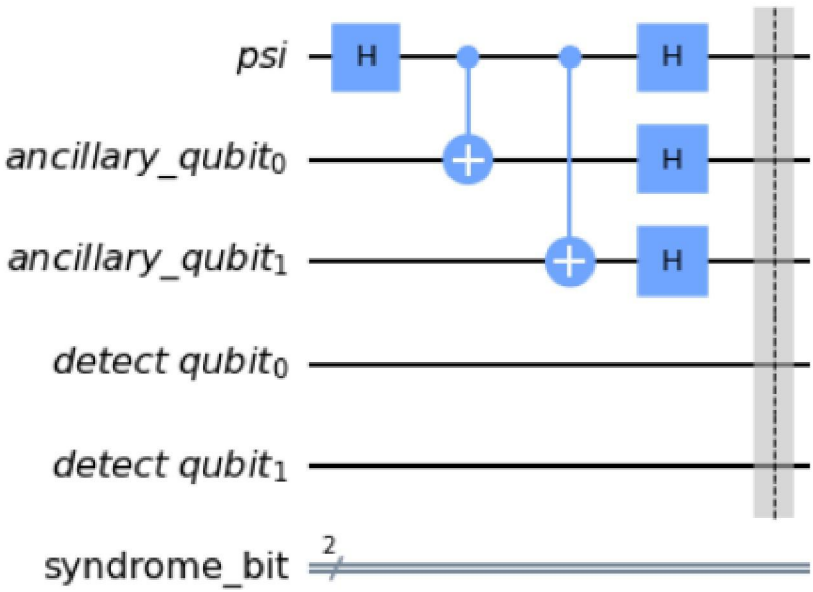


FIG.4. The encoding part prepares the initial state to deal with phase flip errors. $|\psi\rangle = a|+++\rangle + b|---\rangle$

We assume that the phase of the 2nd qubit changes to the opposite.

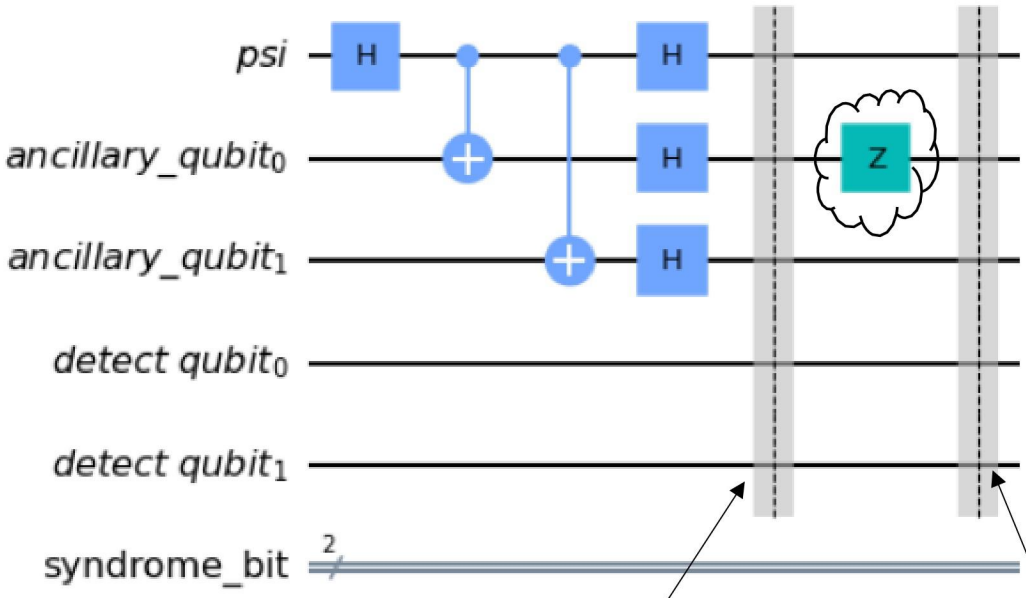
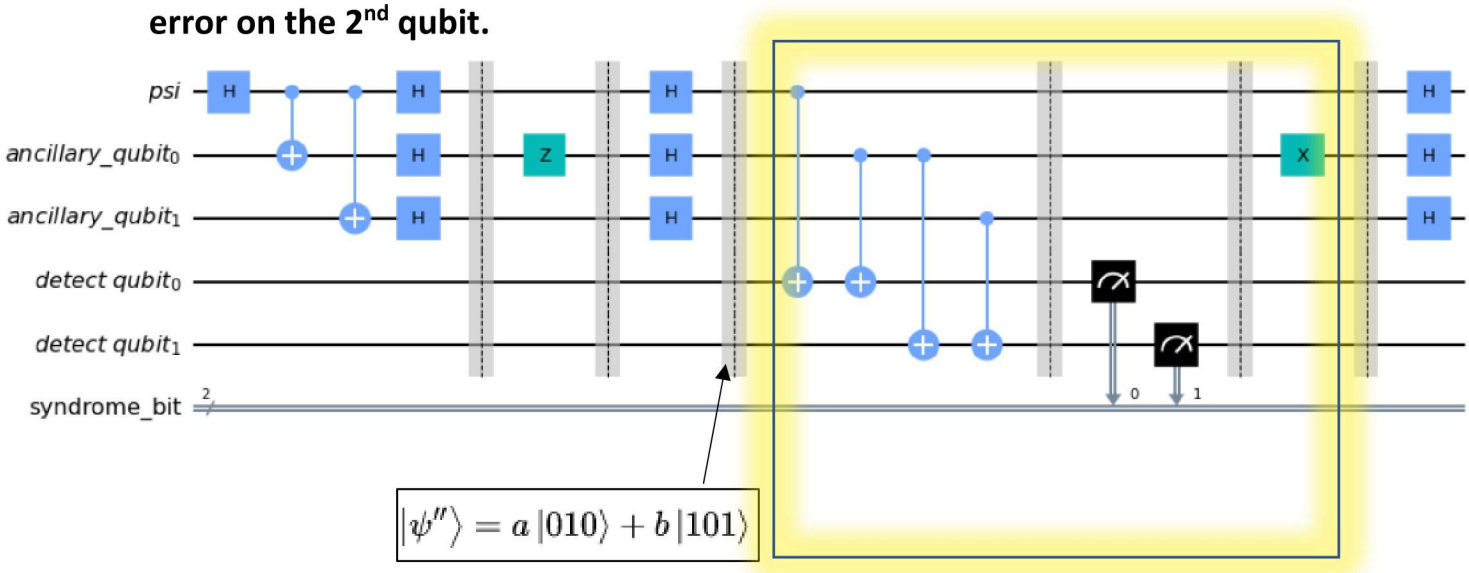


FIG.5. Error on the 2nd qubit

Now in a real circuit **by applying *H* gates we return to the basis of** $|0\rangle$, $|1\rangle$

And we encounter the error we the same procedure as it was a bit flip error on the 2nd qubit.



The highlighted part is implemented from the 3-qubit bit flip code and we can see that the error corrected by applying a *X* gate at the 2nd qubit.

$$|\psi_{01Corrected}\rangle = a|000\rangle + b|111\rangle \longrightarrow |\psi_{+-Corrected}\rangle = a|+++\rangle + b|---\rangle$$

So we prove that the *3-qubit phase flip code* works and the basic **3-qubit bit flip code is fundamental** for other error correcting codes!

Stabilizers, Shor's code and CSS codes (Calderbank, Shor, Steane)

Shor's 9-qubit code

This code can protect the qubit from the effects of arbitrary error, even if it is phase flip or bit flip error! That's why the code is a combination of the 3-qubit phase flip and bit flip codes. Initially the encoding follows the same steps with the phase flip encoding :

$$|\psi\rangle = a|0\rangle + b|1\rangle \longrightarrow a|000\rangle + b|111\rangle \longrightarrow a|+++\rangle + b|---\rangle$$

The next step is to encode each one of the 3 qubits is entagled with 2 ancillary qubits with CNOT gates, like in the bit flip code, so the vectors that form the basis are modified.

$$CNOTCNOT(|+\rangle|0\rangle|0\rangle) = CNOTCNOT(\frac{|0\rangle + |1\rangle}{\sqrt{2}}|00\rangle) =$$

$$= \frac{|000\rangle + |111\rangle}{\sqrt{2}} \text{ , both CNOT gates have the 1st qubit as control qubit}$$

$$CNOTCNOT(|-\rangle|0\rangle|0\rangle) = CNOTCNOT(\frac{|0\rangle - |1\rangle}{\sqrt{2}}|00\rangle) =$$

$$= \frac{|000\rangle - |111\rangle}{\sqrt{2}} \text{ , both CNOT gates have the 1st qubit as control qubit}$$

The result is a 9-qubit encoding with the basis:

$$|0_L\rangle = \frac{(|000\rangle + |111\rangle)(|000\rangle + |111\rangle)(|000\rangle + |111\rangle)}{2\sqrt{2}}$$

$$|1_L\rangle = \frac{(|000\rangle - |111\rangle)(|000\rangle - |111\rangle)(|000\rangle - |111\rangle)}{2\sqrt{2}}$$

And the after the encoding the state is :

$$|\psi\rangle = a\frac{(|000\rangle + |111\rangle)(|000\rangle + |111\rangle)(|000\rangle + |111\rangle)}{2\sqrt{2}} + b\frac{(|000\rangle - |111\rangle)(|000\rangle - |111\rangle)(|000\rangle - |111\rangle)}{2\sqrt{2}}$$

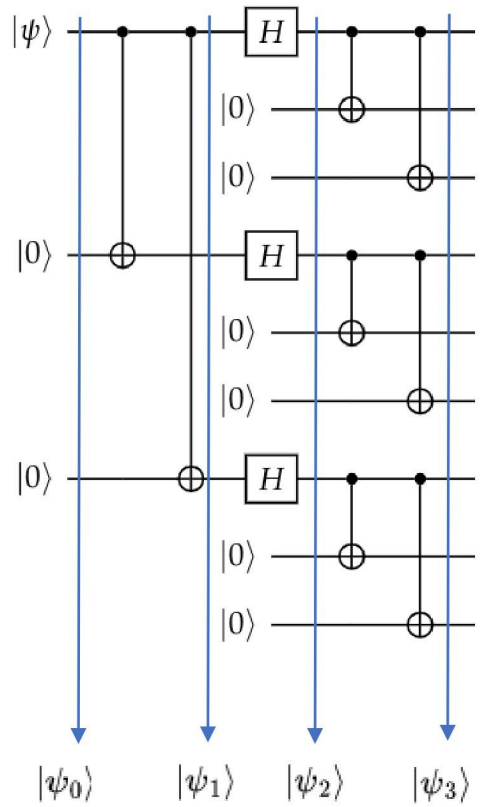


FIG.6. Quantum state evolution

$$|\psi_0\rangle = |\psi 00\rangle \longrightarrow |\psi_1\rangle = a\,|000\rangle + b\,|111\rangle \longrightarrow |\psi_2\rangle = a\,|+++\rangle + b\,|---\rangle$$
$$\rightarrow |\psi_3\rangle = a(\frac{|000\rangle + |111\rangle}{\sqrt{2}})^{\otimes 3} + b(\frac{|000\rangle - |111\rangle}{\sqrt{2}})^{\otimes 3}$$

Bit flip syndromes

The 9-qubit code detects a bit flip error on a single qubit, by performing the operators $Z_1Z_2 = Z \otimes I^{\otimes 2} \otimes Z \otimes I^{\otimes 6}$, comparing the first two qubits to see if they are different. Then we compare the 2nd and the 3rd qubit to see if they are different, $Z_2Z_3 = I^{\otimes 3} \otimes Z \otimes I^{\otimes 3} \otimes Z \otimes I^{\otimes 3}$

The results are calculated from the measurements:

$M_1 = \langle\psi|Z_1Z_2\,|\psi\rangle\ ,\ M_2 = \langle\psi|Z_2Z_3\,|\psi\rangle$

In reality these are stabilizers, more information later!

The possible error syndromes are:

$M_1 = -1$	$M_2 = 1$: bit flip error on 1st qubit
$M_1 = -1$	$M_2 = -1$: bit flip error on 2nd qubit
$M_1 = 1$	$M_2 = -1$: bit flip error on 2nd qubit

(6)

We recover from this type of error by flipping the wrong qubit with a *X* gate**. Correction operators** -> **(2)**

Let's take for example that **the first bit flipped**. For bit flip error we only need to word with the first three qubits a not with the whole nine qubit string.

$$|\psi\rangle = a\frac{(|000\rangle + |111\rangle)(|000\rangle + |111\rangle)(|000\rangle + |111\rangle)}{2\sqrt{2}} + b\frac{(|000\rangle - |111\rangle)(|000\rangle - |111\rangle)(|000\rangle - |111\rangle)}{2\sqrt{2}}$$

$$|\psi\rangle = a\frac{|000\rangle + |111\rangle}{2\sqrt{2}} + b\frac{|000\rangle - |111\rangle}{2\sqrt{2}} \longrightarrow error$$

$$\longrightarrow |\psi'\rangle = a\frac{|100\rangle + |011\rangle}{2\sqrt{2}} + b\frac{|100\rangle - |011\rangle}{2\sqrt{2}}$$

$$Z_1 Z_2 |\psi'\rangle = -a\frac{|100\rangle + |011\rangle}{2\sqrt{2}} - b\frac{|100\rangle - |011\rangle}{2\sqrt{2}} = -|\psi'\rangle$$

$$M_1 = \langle\psi'|Z_1 Z_2|\psi'\rangle = -\langle\psi'|\psi'\rangle = -1$$

$$Z_2 Z_3 |\psi'\rangle = a\frac{|100\rangle + |011\rangle}{2\sqrt{2}} + b\frac{|100\rangle - |011\rangle}{2\sqrt{2}} = |\psi'\rangle$$

$$M_2 = \langle\psi'|Z_1 Z_2|\psi'\rangle = \langle\psi'|\psi'\rangle = 1$$

(6) The system detected bit flip error on the first qubit. The correction procedure will flip the 1st bit (2).

Phase flip syndromes

A phase flip error whenever it occurs on any qubit out of three, it flips the sign of the block the qubit belongs to. Namely $\frac{|000\rangle + |111\rangle}{\sqrt{2}} \longrightarrow \frac{|000\rangle - |111\rangle}{\sqrt{2}}$ and vice versa. The 9-qubit code detects a phase flip error by comparing the sign of the 1st and 2nd block and the sign of the 2nd and 3rd block. The result of each comparison reflects the result of the measurement

$$M_1 = \langle\psi|X^{\otimes 6} \otimes I^{\otimes 3}|\psi\rangle \qquad M_2 = \langle\psi|I^{\otimes 3} \otimes X^{\otimes 6}|\psi\rangle$$

$$\uparrow \qquad\qquad\qquad \uparrow$$

$$X_1X_2X_3X_4X_5X_6 \qquad\qquad X_4X_5X_6X_7X_8X_9$$

Then according to the measurement results we apply a *Z* gate to flip the qubit's phase.

$M_1 = -1$	$M_2 = 1$	: Z on 1st qubit
$M_1 = -1$	$M_2 = -1$	: Z on 2nd qubit
$M_1 = 1$	$M_2 = -1$	: Z on 3rd qubit

(7)

Let's see an example. Suppose a phase flip error occurs on the 1st qubit. That means the sign of the 1st block changes to the opposite.

$$|\psi\rangle = a\frac{(|000\rangle + |111\rangle)(|000\rangle + |111\rangle)(|000\rangle + |111\rangle)}{2\sqrt{2}} + b\frac{(|000\rangle - |111\rangle)(|000\rangle - |111\rangle)(|000\rangle - |111\rangle)}{2\sqrt{2}} \longrightarrow error$$

$$|\psi'\rangle = a\frac{(Z|0\rangle|00\rangle + Z|1\rangle|11\rangle)(|000\rangle + |111\rangle)(|000\rangle + |111\rangle)}{2\sqrt{2}} + b\frac{(Z|0\rangle|00\rangle - Z|1\rangle|11\rangle)(|000\rangle - |111\rangle)(|000\rangle - |111\rangle)}{2\sqrt{2}}$$

$$|\psi'\rangle = a\frac{(|000\rangle - |111\rangle)(|000\rangle + |111\rangle)(|000\rangle + |111\rangle)}{2\sqrt{2}} + b\frac{(|000\rangle + |111\rangle)(|000\rangle - |111\rangle)(|000\rangle - |111\rangle)}{2\sqrt{2}}$$

Now we will perform the measurements:

$$X_1X_2X_3X_4X_5X_6|\psi'\rangle = a\frac{(|111\rangle - |000\rangle)(|111\rangle + |000\rangle)(|000\rangle + |111\rangle)}{2\sqrt{2}} + b\frac{(|111\rangle + |000\rangle)(|111\rangle - |000\rangle)(|000\rangle - |111\rangle)}{2\sqrt{2}} =$$

$$= -|\psi'\rangle$$

$$M_1 = -\langle\psi'|\psi'\rangle = -1$$

$$X_4X_5X_6X_7X_8X_9|\psi'\rangle = a\frac{(|000\rangle - |111\rangle)(|111\rangle + |000\rangle)(|111\rangle + |000\rangle)}{2\sqrt{2}} + b\frac{(|000\rangle + |111\rangle)(|111\rangle - |000\rangle)(|111\rangle - |000\rangle)}{2\sqrt{2}} =$$

$$= |\psi'\rangle$$

$$M_1 = \langle\psi'|\psi'\rangle = 1$$

(7) The system detected the phase flip error on the first block and will perform the recovery according to the operators at (5).

Circuit representation

the circuit consists of the encoding part, the error detection part and the error correction part.

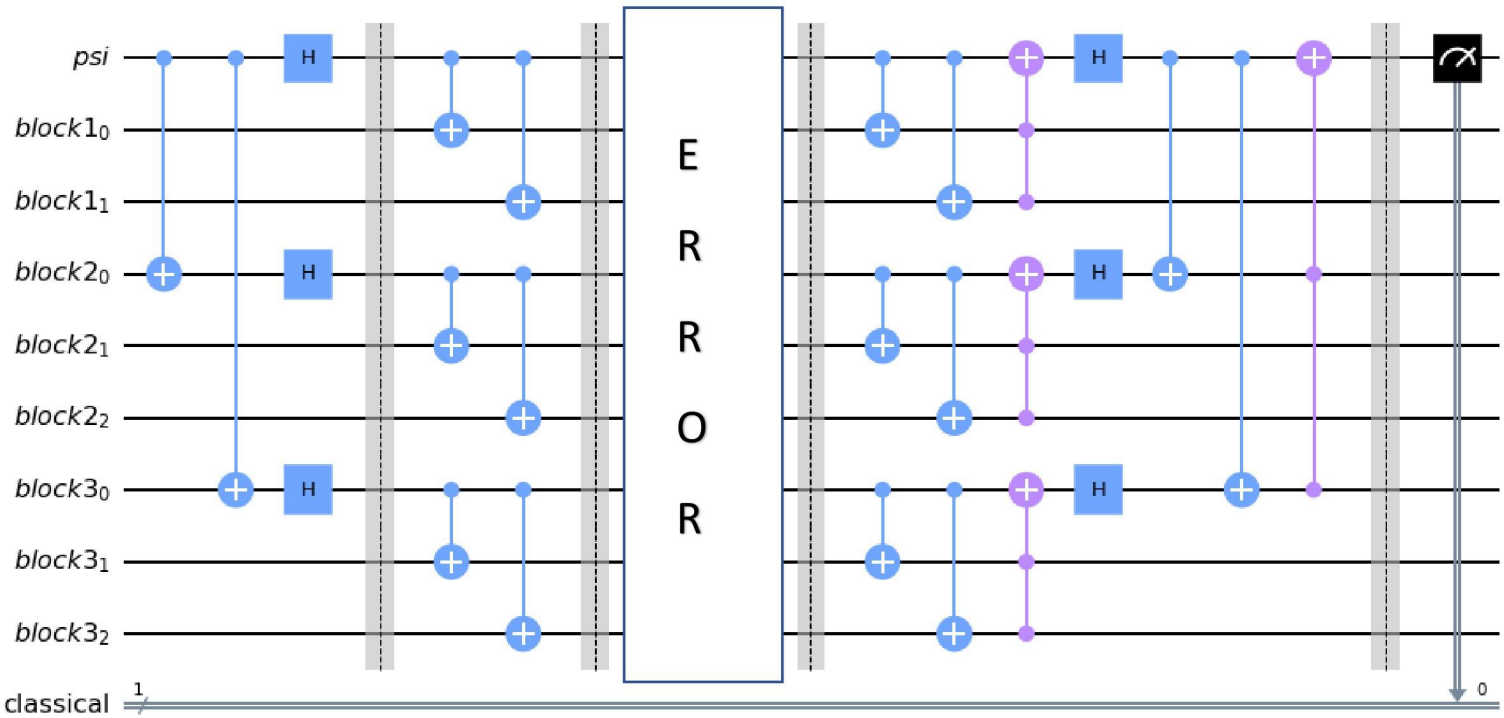


FIG.7. shor's code circuit

The 2nd and the 6th qubit are responsible for bit flip correction. The auxiliary qubits are responsible for phase correction. The last step of the encoding is repeated and then Toffoli gates are applied to the main qubit as well as to the 3rd and the 6th. The control qubits are the auxiliary qubits responsible for *phase flip correction.*

Hadamard gates are applied to bring the 3 qubits out of the superposition. Then CNOT gates are applied to the 3rd and 6th qubit where the control qubit is the main qubit. Finally a Toffoli gate is applied to the main qubit which is controlled by the 3rd and 6th qubit.

Let's take for example the initial state $|\psi\rangle = \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$ and assume that a bit flip and phase flip error occurs on the 1st qubit.

The error operators are *X and Z.*

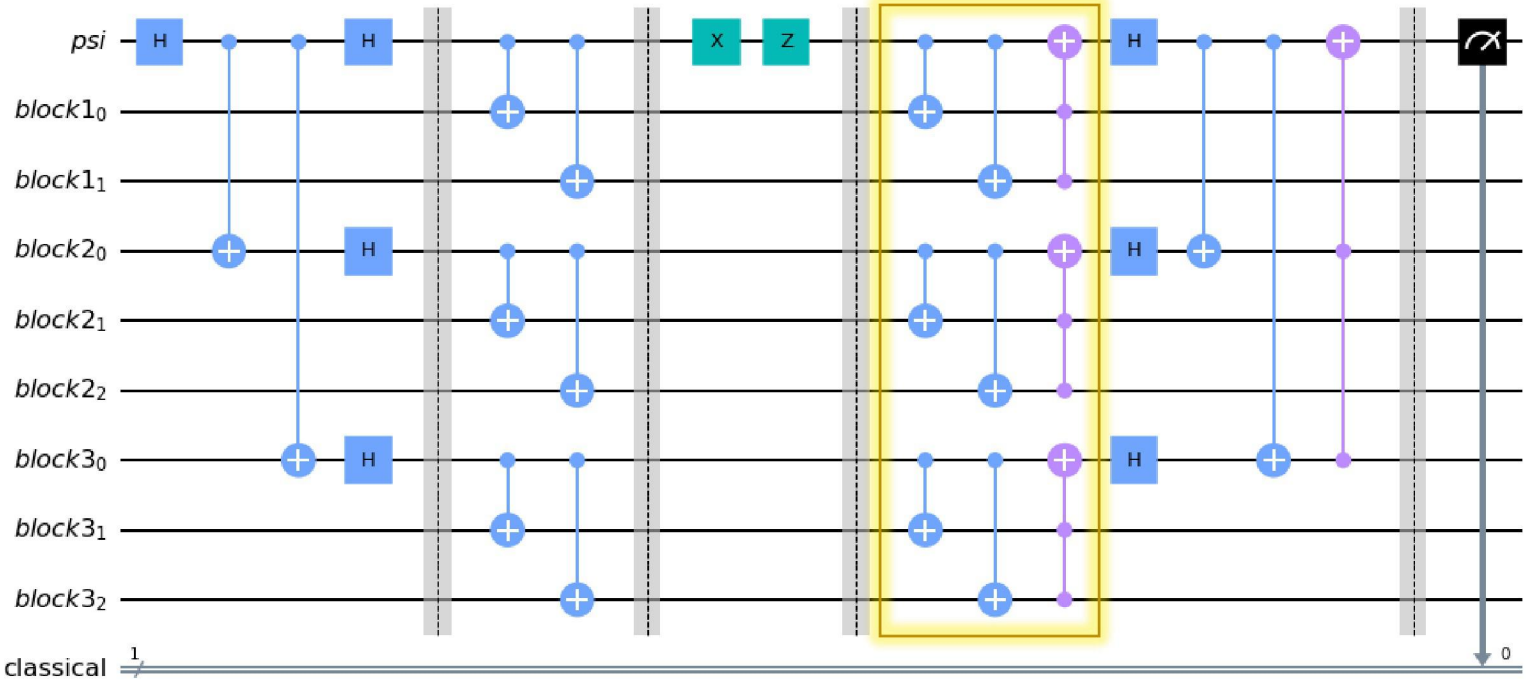


FIG.8.

$$|\psi\rangle = a(\frac{|000\rangle + |111\rangle}{\sqrt{2}})^{\otimes 3} + b(\frac{|000\rangle - |111\rangle}{\sqrt{2}})^{\otimes 3} \longrightarrow$$

$$|\psi'\rangle = a\frac{(|100\rangle - |011\rangle)(|000\rangle + |111\rangle)(|000\rangle + |111\rangle)}{2\sqrt{2}} + b\frac{(|100\rangle + |011\rangle)(|000\rangle - |111\rangle)(|000\rangle - |111\rangle)}{2\sqrt{2}}$$

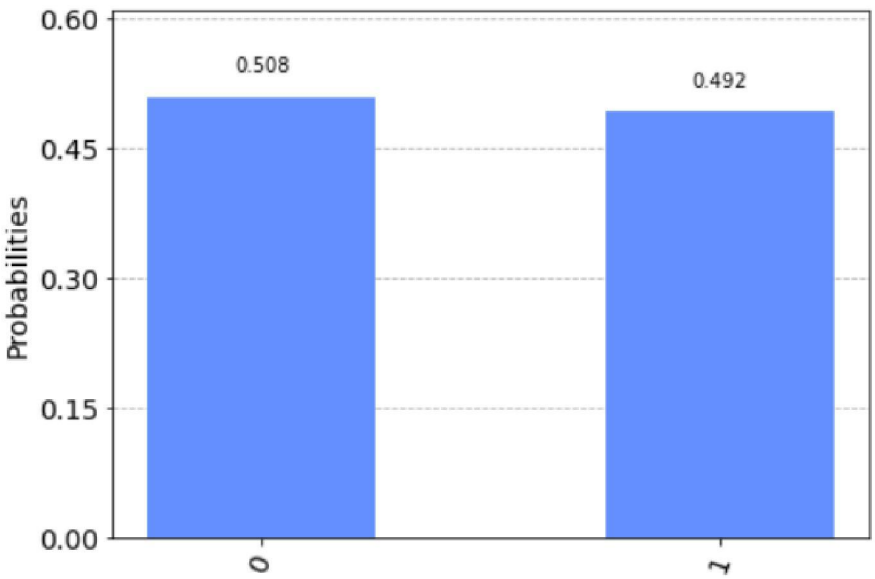
After the set of Cnot and toffoli gates in the highlighted part:

$$|\psi'\rangle = a\frac{(|011\rangle - |111\rangle)(|000\rangle + |100\rangle)(|000\rangle + |100\rangle)}{2\sqrt{2}} + b\frac{(|011\rangle + |111\rangle)(|000\rangle - |100\rangle)(|000\rangle - |100\rangle)}{2\sqrt{2}}$$

After that we will continue the process with the 3 primary qubits.

$|\psi'\rangle = a\,|-++\rangle + b\,|+--\rangle$ then we apply the *H* gates.

$|\psi'\rangle = a\,|100\rangle + b\,|011\rangle$ and applying the last cnot, toffoli gates we measure the final state $|\psi_{final}\rangle = a\,|0\rangle + b\,|1\rangle$, a = b = $\frac{1}{\sqrt{2}}$



PLOT 1

Stabilizer error correcting codes

Many quantum error correcting codes (CSS codes, Shor's code etc) can be much more compactly described using *stabilizers* than in the state vector description. The *stabilizer formalism* needs to be introduced first in order to understand the stabilizers. For instance suppose we have the state $|\psi\rangle = \frac{|00\rangle + |11\rangle}{\sqrt{2}}$ which is the Bell state $|\beta_{00}\rangle$.

It is simple to verify that $X \otimes X\,|\psi\rangle = X1X2\,|\psi\rangle = |\psi\rangle$ and $Z \otimes Z\,|\psi\rangle = Z1Z2\,|\psi\rangle = |\psi\rangle$. That means ***the state is stabilized*** bv the operators $X1X2$ and $Z1Z2$. It's also interesting that the state $|\psi\rangle$ is the unique state which is stabilized by these two operators.

The stabilizer formalism originates from the use of *group theory*. Consider the group that contains all the Pauli operators with multiplicative factors $\pm i, \pm 1$:

$$G = \{\pm I, \pm iI, \pm X, \pm iX, \pm Y, \pm iY, \pm Z, \pm iZ\}$$

For an error correcting algorithm instead of the projection operators that detect a qubit suppose we have a subspace S of the group G. The description of the group S could be approached accurately by its ***generators***. Generators is a set of elements $g_1 \dots g_n$ that can generate the group G, if every element of G can be written as a product of elements from the list $g_1 \dots g_n$ and we write -> $G =< g_1 \dots g_n >$.

If we review the previous example, for $|\psi\rangle = \frac{|00\rangle + |11\rangle}{\sqrt{2}}$ we could come up with the stabilizer generators

$$G =< g_1 g_2 >, \; g_1 = X_1 X_2 \quad and \quad g_2 = Z_1 Z_2$$

Note: the stabilizer formalism states that $X_i X_j = X_i \otimes X_j$, i and j, are random indexes.

The 9-qubit Shor's code we explained before could be encoded with the use of its own generators. Each generator if it operates on

$$|0_L\rangle = \frac{(|000\rangle + |111\rangle)(|000\rangle + |111\rangle)(|000\rangle + |111\rangle)}{2\sqrt{2}}$$

the result will be $|0_L\rangle$ again.

The 8 generators are:

Name	Operator
g_1	$ZZIIIIIII$
g_2	$IZZIIIIII$
g_3	$IIIIZZIII$
g_4	$IIIIIIZZI$
g_5	$IIIIIIIZZ$
g_6	$XXXXXXIII$
g_7	$IIIXXXXXX$
g_8	

(8)

For instance:

$$g_8 |0_L\rangle = g_8 \bullet (|000000000\rangle + |000000111\rangle + |000111000\rangle + |000111111\rangle + \\ + |111000000\rangle + |111000111\rangle + |111111000\rangle + |111111111\rangle) =$$

$$= |000111111\rangle + |000111000\rangle + |000000111\rangle + |000000000\rangle + \\ + |111111111\rangle + |111111000\rangle + |111000111\rangle + |111000000\rangle =$$

$$= |0_L\rangle$$

7-qubit Steane code

This error correction code uses 7 qubits for the encoding and the *logical qubits* $|0_L\rangle,\ |1_L\rangle$ are estimated according to the stabilizer generators.

The 6 generators for the 7-qubit code are:

Name	Operator
g_1	$I\ I\ I\ X\ X\ X\ X$
g_2	$I\ X\ X\ I\ I\ X\ X$
g_3	$X\ I\ X\ I\ X\ I\ X$
g_4	$I\ I\ I\ Z\ Z\ Z\ Z$
g_5	$I\ Z\ Z\ I\ I\ Z\ Z$
g_6	$Z\ I\ Z\ I\ Z\ I\ Z$

(9)

Th logical states of the Steane code are:

$$|0\rangle_L = \frac{1}{\sqrt{8}}[|0000000\rangle + |1010101\rangle + |0110011\rangle + |1100110\rangle + |0001111\rangle + |1011010\rangle + |0111100\rangle + |1101001\rangle]$$

$$|1\rangle_L = X_L|0\rangle_L.$$

The first three generators g_1, g_2, g_3 operate as phase flip detectors.

The last three generators g_4, g_5, g_6 operate as bit flip detectors.

This particular code is somewhat a copy of the **Hamming code** that uses an extra parity bit**,** which is used in classical computers and it also does parity check.

$$\mathbf{G} := \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \end{pmatrix}_{4,8}$$

The encoding for example of the message x = (1 0 1 1) is performed with the generator matrix:

$$encode = xG = (1\,0\,1\,1)\begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \end{pmatrix} = (2 \quad 3 \quad 1 \quad 2 \quad 0 \quad 1 \quad 1 \quad 2) mod 2 = 0\,1\,1\,0\,0\,1\,1\,0$$

Circuit representation 7-qubit Steane Code

Let's see how a real circuit encodes an arbitrary state.

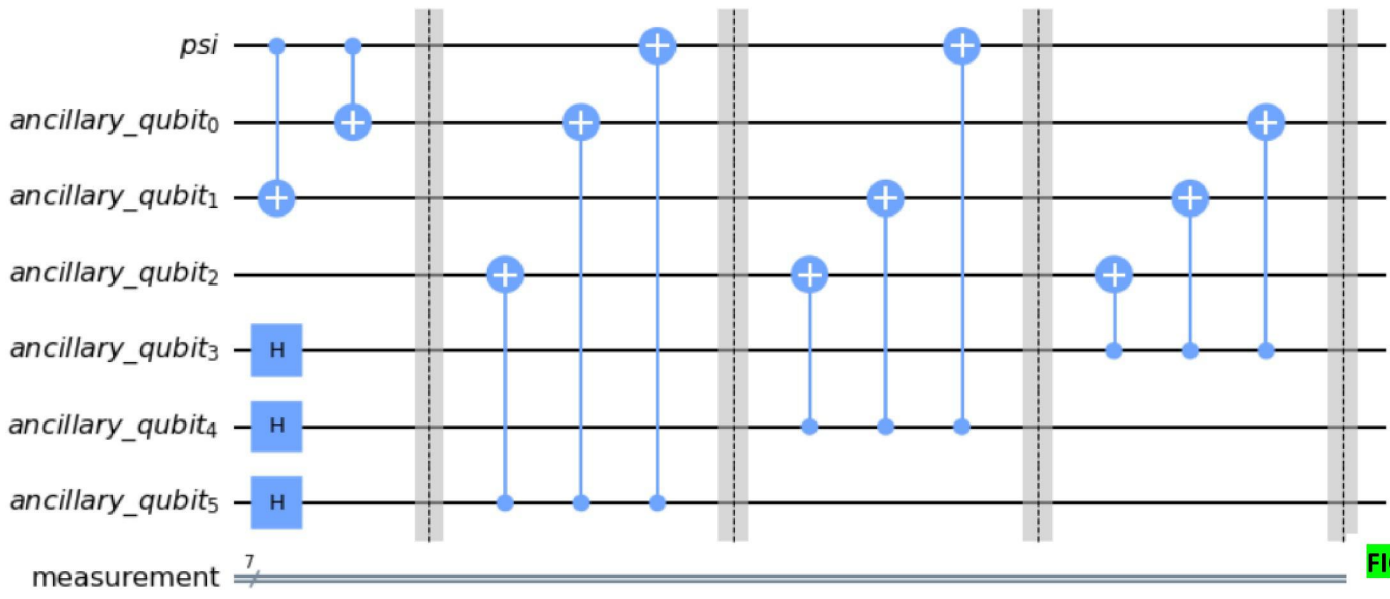
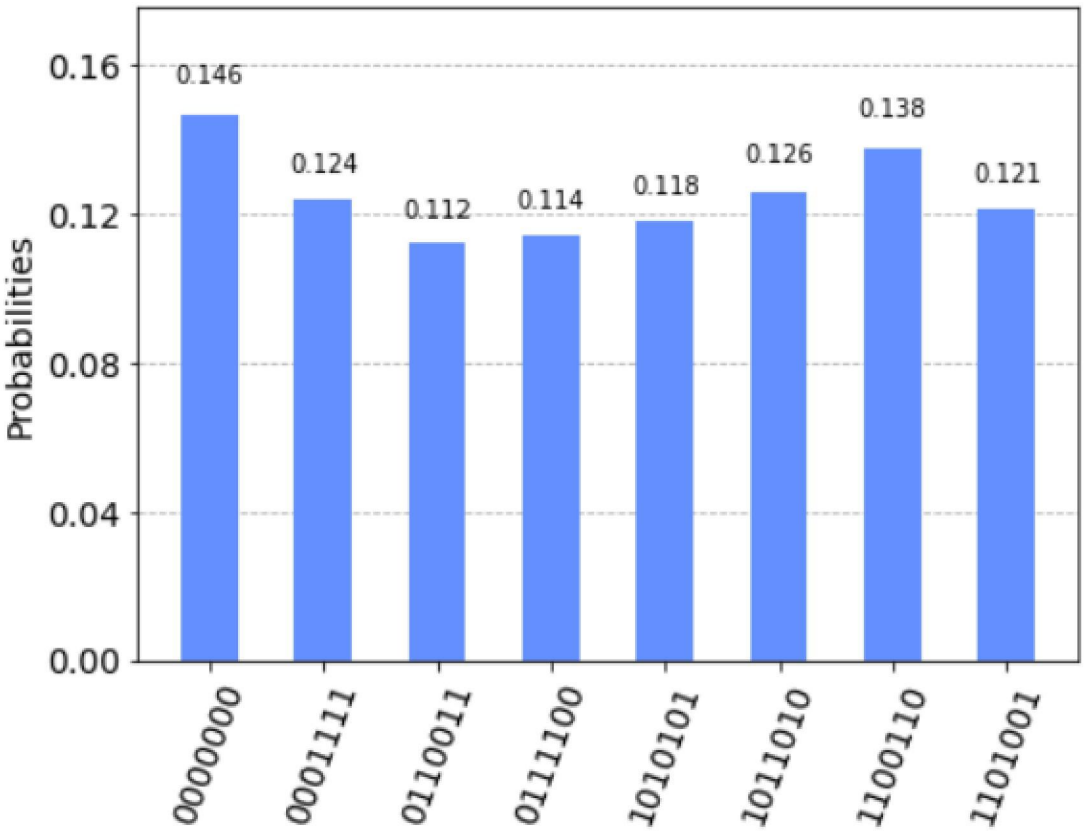


FIG.9.

This circuit can encode any state $|\psi\rangle = a\,|0\rangle + b\,|1\rangle$. We will see how the logical state $|0_L\rangle$ is encoded according to this circuit. Consider $|psi\rangle = |0\rangle$

If we measure the 7 qubits the results are:



PLOT 2

Each element is shown with probability approximately $(\frac{1}{\sqrt{8}})^2 = 0.125$ and it is easy to observe that the histogram matches the state

$$|0\rangle_L = \frac{1}{\sqrt{8}}[|0000000\rangle + |1010101\rangle + |0110011\rangle + |1100110\rangle + |0001111\rangle + |1011010\rangle + |0111100\rangle + |1101001\rangle]$$

Thus it is clear that the circuit encodes any arbitrary correctly to:

$$|\psi\rangle = a\,|0_L\rangle + b\,|1_L\rangle$$

$$|\psi\rangle = a\frac{1}{\sqrt{8}}[|0000000\rangle + |1010101\rangle + |0110011\rangle + |1100110\rangle + |0001111\rangle + |1011010\rangle + |0111100\rangle + |1101001\rangle] + bX_L|0\rangle_L.$$

5-qubit code

Now that we have understood the stabilizers and the stabilizer formalism, we can introduce the *5-qubit code*, which is the smallest quantum error correcting code that can protect a logical qubit against an arbitrary error on a single qubit. The initial state is encoded to 5 physical qubits and at the end of the encoding the logical qubits are stabilized by the code's **generators**.

The generators are:

Name	Operator
g_1	$XZZXI$
g_2	$IXZZX$
g_3	$XIXZZ$
g_4	$ZXIXZ$

(10)

- The 1st stabilizer does a parity measurement of bit flip error on the 1st qubit,
- the 2nd stabilizer checks for phase flip error on the 2nd qubit,
- the 3rd checks for phase flip error on the 3rd and
- the 4th checks for bit flip error on the 4th qubit.

For many applications it is more transparent to use the **7-qubit Steane code**, but the **5-qubit code** it's very useful because it has the smallest number of qubits and corrects errors on a single qubit perfectly!

Circuit of 5-qubit code

The circuit encodes the logical qubits into a superposition of 5 physical qubits. The logical qubits that are stabilized by the generators are:

$$|0_L\rangle = \frac{1}{4}(+|00000\rangle - |00011\rangle + |00101\rangle - |00110\rangle + |01001\rangle + |01010\rangle - |01100\rangle - |01111\rangle$$
$$- |10001\rangle + |10010\rangle + |10100\rangle - |10111\rangle - |11000\rangle - |11011\rangle - |11101\rangle - |11110\rangle)$$

$$|1_L\rangle = \frac{1}{4}(+|11111\rangle - |11100\rangle + |11010\rangle - |11001\rangle + |10110\rangle + |10101\rangle - |10011\rangle - |10000\rangle$$
$$- |01000\rangle + |01011\rangle + |01101\rangle - |01110\rangle - |00001\rangle - |00010\rangle - |00100\rangle - |00111\rangle)$$

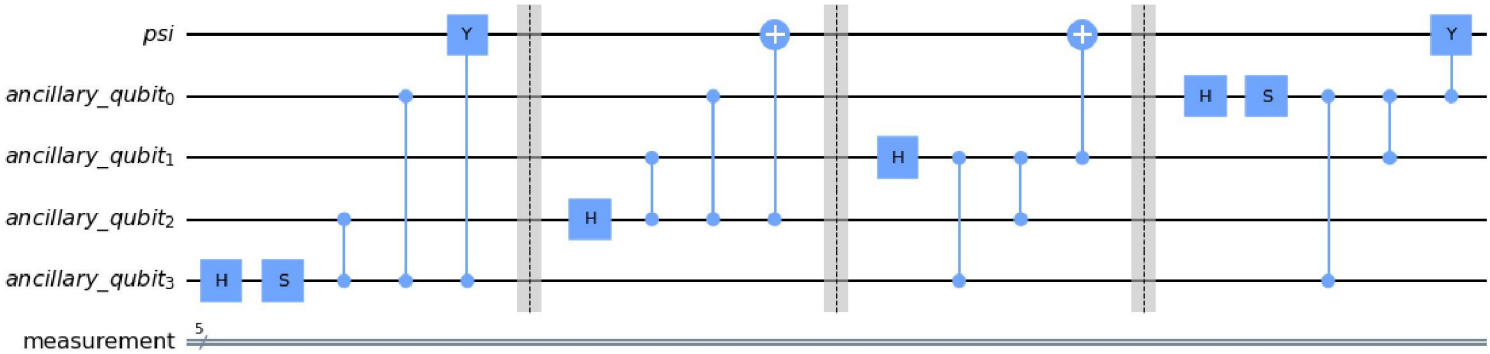
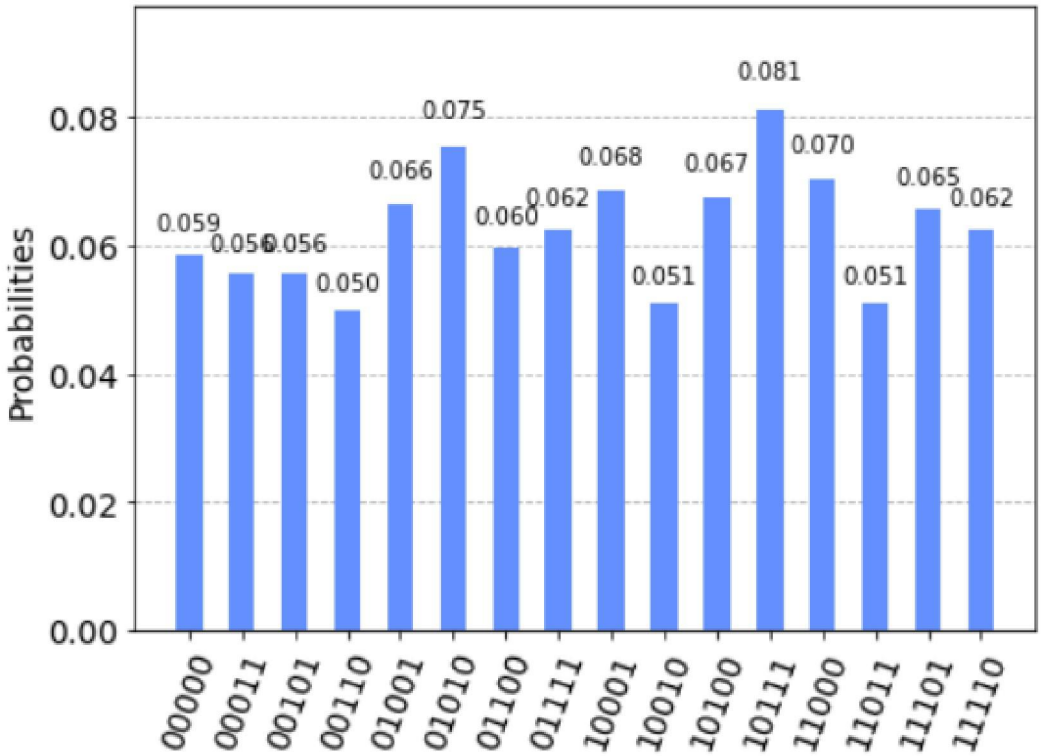


FIG.10. Circuit represantation of the 5-qubit code

In order to understand how the logical qubits are encoded, suppose again $|psi\rangle = |0\rangle$ and then we measure all the qubits. The result should be a superposition of 5 qubits:

$$|0_L\rangle = \frac{1}{4}(+|00000\rangle - |00011\rangle + |00101\rangle - |00110\rangle + |01001\rangle + |01010\rangle - |01100\rangle - |01111\rangle$$
$$- |10001\rangle + |10010\rangle + |10100\rangle - |10111\rangle - |11000\rangle - |11011\rangle - |11101\rangle - |11110\rangle)$$

After the measurement:



PLOT 3

We see that each 5 qubit string is what we expected and the probability of each element is close to $(\frac{1}{4})^2 = 0.625$

Qiskit executable code(ipynb):

https://drive.google.com/file/d/1Dr60sGlkSehxZeSDjZ3v4DsGgeCWtlN1/view?usp=share_link

Sources

Nielsen & Chuang Book:
https://www.eclass.tuc.gr/modules/document/file.php/HMMY256/%CE%92%CE%B9%CE%B2%CE%BB%CE%AF%CE%B1/Nielsen%26Chuang.pdf

Oskin's notes:
https://www.eclass.tuc.gr/modules/document/file.php/HMMY256/Oskin_lecture_notes_Washington.pdf

Circuit diagram of Shor's code:

https://quantumcomputinguk.org/tutorials/quantum-error-correction-shor-code-in-qiskit

Steane code logical states:

https://en.wikipedia.org/wiki/Steane_code

5-qubit code, encoding and logical states:

https://quantumcomputing.stackexchange.com/questions/14264/nielsenchuang-5-qubit-quantum-error-correction-encoding-gate

Functions editor:

https://latexeditor.lagrida.com/

Quantum Error Correction

ECE TUC Autumn 2022 Introduction to Quantum Computers Course

Athanasios Karakos


```
from qiskit.providers.aer.noise import NoiseModel
from qiskit.providers.aer.noise.errors import pauli_error,
depolarizing_error
from qiskit import QuantumRegister, ClassicalRegister
from qiskit import QuantumCircuit, Aer, transpile, assemble
from qiskit.visualization import plot_bloch_multivector,
plot_histogram, array_to_latex
from qiskit.quantum_info import Statevector
from numpy import pi
import random
```

3-QUBIT BIT FLIP CODE

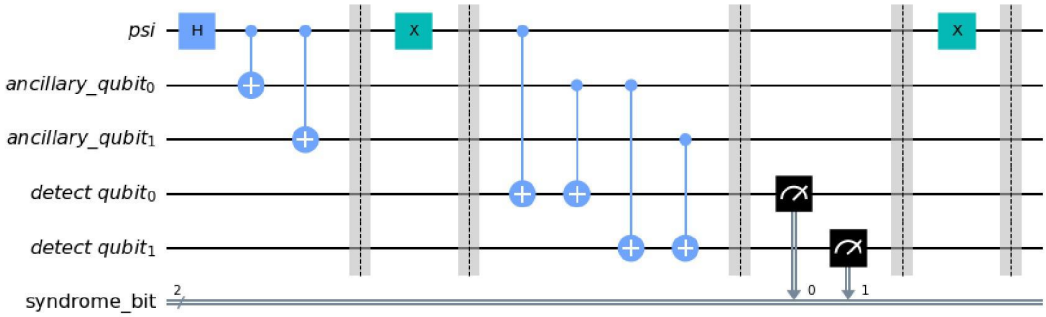
```
def bit_flip_encode(qc):
    qc.cx(init[0],aq[0])
    qc.cx(init[0],aq[1])
    qc.barrier()
    return qc
def error_area(qc):
    p = 0.4
    pc = random.random()
    if pc < p:
        qc.x(random.choice([aq[0],aq[1],init[0]]))
        qc.barrier()
    return qc
def bit_flip_syndrome(qc):
    qc.cx(init[0],sq[0])
    qc.cx(aq[0],sq[0])
    qc.cx(aq[0],sq[1])
    qc.cx(aq[1],sq[1])
    qc.barrier()
    qc.measure(sq[0],sb[0])
    qc.measure(sq[1],sb[1])
    qc.barrier()
    return qc
def bit_flip_correct(qc):
    ##########################################
    zeros = QuantumCircuit(2,2)
    zeros.measure(0,0)
    zeros.measure(1,1)
    qobj = assemble(zeros)
    count00 = sim.run(qobj).result().get_counts()
    ##########################################
```

```
    zerone = QuantumCircuit(2,2)
    zerone.x(1)
    zerone.measure(0,0)
    zerone.measure(1,1)
    qobj = assemble(zerone)
    count01 = sim.run(qobj).result().get_counts()
    ##########################################
    onezero = QuantumCircuit(2,2)
    onezero.x(0)
    onezero.measure(0,0)
    onezero.measure(1,1)
    qobj = assemble(onezero)
    count10 = sim.run(qobj).result().get_counts()
    ##########################################
    ones = QuantumCircuit(2,2)
    ones.x(0)
    ones.x(1)
    ones.measure(0,0)
    ones.measure(1,1)
    qobj = assemble(ones)
    count11 = sim.run(qobj).result().get_counts()
    ##########################################
    temp = qc
    qobj = assemble(temp)
    counts = sim.run(qobj).result().get_counts()
    if counts == count00:
        pass
    elif counts == count01:
        qc.x(aq[1])
        qc.barrier()
    elif counts == count10:
        qc.x(init[0])
        qc.barrier()
    elif counts == count11:
        qc.x(aq[0])
        qc.barrier()
    else:
        pass

    return qc

aq = QuantumRegister(2,'ancillary_qubit')
sb = ClassicalRegister(2,'syndrome_bit')
sq = QuantumRegister(2,'detect qubit')
init = QuantumRegister(1,'psi')
qc = QuantumCircuit(init,aq,sq,sb)
qc.h(init[0])
qc = bit_flip_encode(qc)
qc = error_area(qc)
qc = bit_flip_syndrome(qc)
```

```
qc = bit_flip_correct(qc)
qc.draw(output='mpl')
```

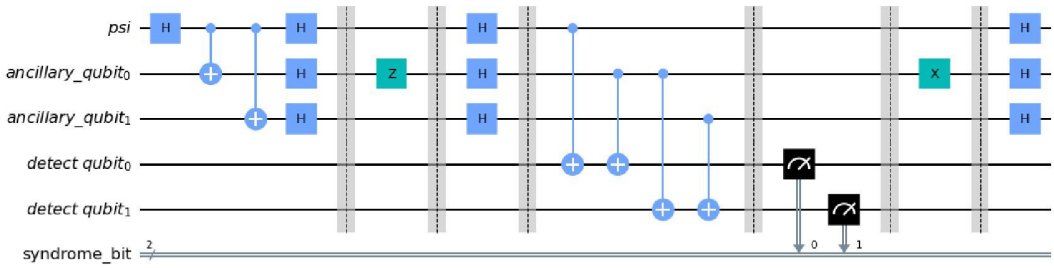


3-QUBIT PHASE FLIP CODE

```
def phase_flip_encode(qc):
    qc.cx(init[0],aq[0])
    qc.cx(init[0],aq[1])
    qc.h(init[0])
    qc.h(aq[0])
    qc.h(aq[1])
    qc.barrier()
    return qc
def error_area(qc):
    p = 0.4
    pc = random.random()
    if pc < p:
        qc.z(random.choice([aq[0],aq[1],init[0]]))
        qc.barrier()
    return qc
def phase_flip_syndrome(qc):
    qc.h(init[0])
    qc.h(aq[0])
    qc.h(aq[1])
    qc.barrier()
    qc.cx(init[0],sq[0])
    qc.cx(aq[0],sq[0])
    qc.cx(aq[0],sq[1])
    qc.cx(aq[1],sq[1])
    qc.barrier()
    qc.measure(sq[0],sb[0])
    qc.measure(sq[1],sb[1])
    qc.barrier()
    return qc

aq = QuantumRegister(2,'ancillary_qubit')
sb = ClassicalRegister(2,'syndrome_bit')
sq = QuantumRegister(2,'detect qubit')
init = QuantumRegister(1,'psi')
```

```
qc = QuantumCircuit(init,aq,sq,sb)
qc.h(init[0])
qc = phase_flip_encode(qc)
qc = error_area(qc)
qc = phase_flip_syndrome(qc)
qc = bit_flip_correct(qc)
qc.h([init[0],aq[0],aq[1]])
qc.draw(output='mpl')
```

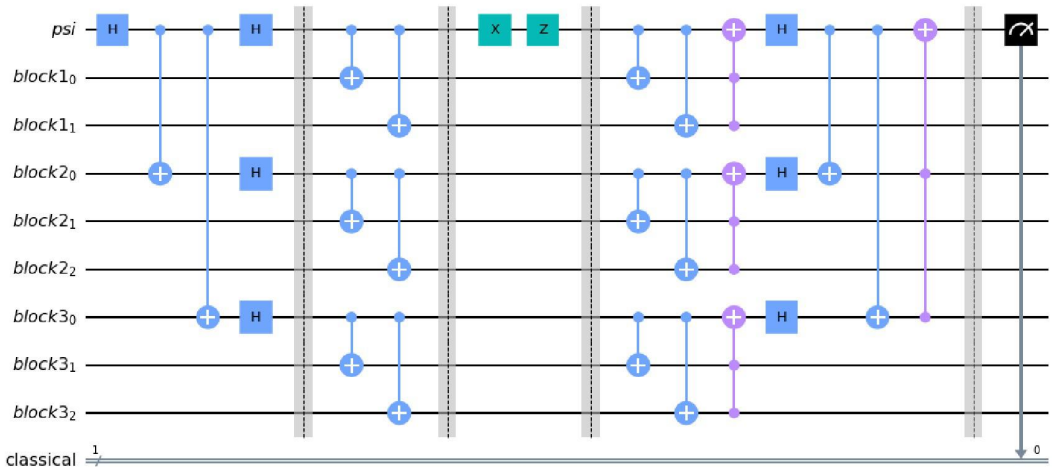


Shor's 9-QUBIT CODE

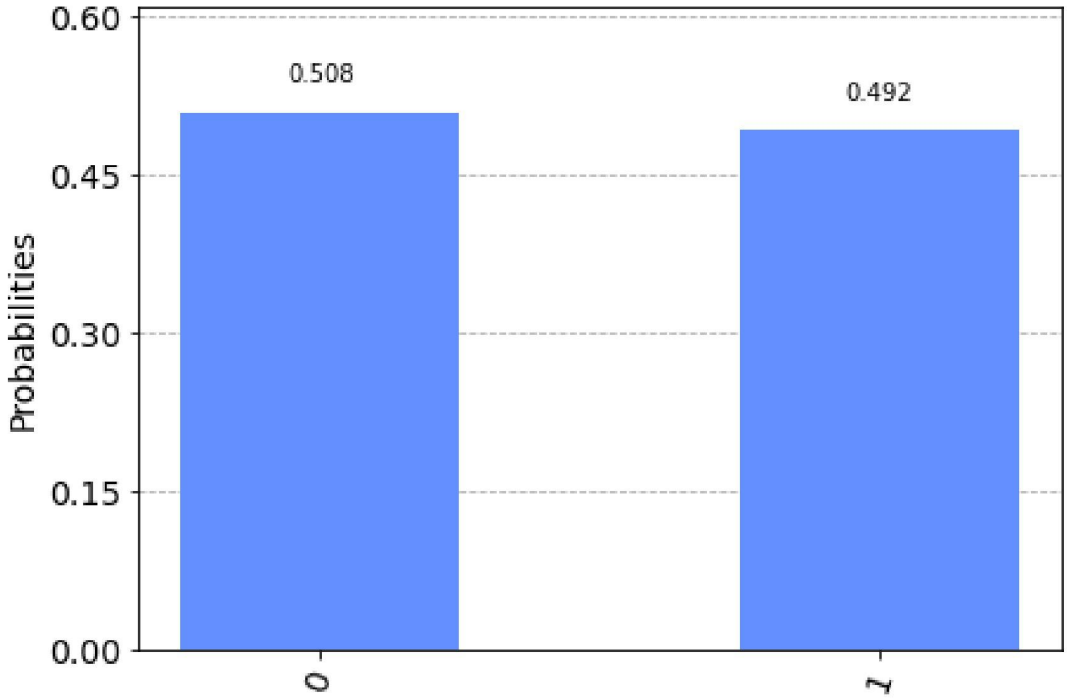
```
def shor_code_encode(qc):
    qc.cx(init[0],block2[0])
    qc.cx(init[0],block3[0])
    qc.h([init[0],block2[0],block3[0]])
    qc.barrier()
    qc.cx(init[0],block1[0])
    qc.cx(init[0],block1[1])
    qc.cx(block2[0],block2[1])
    qc.cx(block2[0],block2[2])
    qc.cx(block3[0],block3[1])
    qc.cx(block3[0],block3[2])
    return qc
def error_area(qc):
    qc.barrier()
    qc.x(0)
    qc.z(0)
    qc.barrier()
    return qc
def error_correction(qc):
    qc.cx(init[0],block1[0])
    qc.cx(init[0],block1[1])
    qc.cx(block2[0],block2[1])
    qc.cx(block2[0],block2[2])
    qc.cx(block3[0],block3[1])
    qc.cx(block3[0],block3[2])
    qc.ccx(block1[1],block1[0],init[0])
    qc.ccx(block2[2],block2[1],block2[0])
    qc.ccx(block3[2],block3[1],block3[0])
    qc.h([init[0],block2[0],block3[0]])
    qc.cx(init[0],block2[0])
```

```
    qc.cx(init[0],block3[0])
    qc.ccx(block3[0],block2[0],init[0])
    return qc


block2 = QuantumRegister(3,'block2')
init = QuantumRegister(1,'psi')
block1 = QuantumRegister(2,'block1')
block3 = QuantumRegister(3,'block3')
cb = ClassicalRegister(1,'classical')
qc = QuantumCircuit(init,block1,block2,block3,cb)
qc.h(0)
qc = shor_code_encode(qc)
qc = error_area(qc)
qc = error_correction(qc)
qc.barrier()
qc.measure(0,0)
qc.draw(output='mpl')
```




```
qobj = assemble(qc)  # Assemble circuit into a Qobj that can be run
counts = sim.run(qobj).result().get_counts()  # Do the simulation,
returning the state vector
plot_histogram(counts)  # Display the output on measurement of state
vector
```

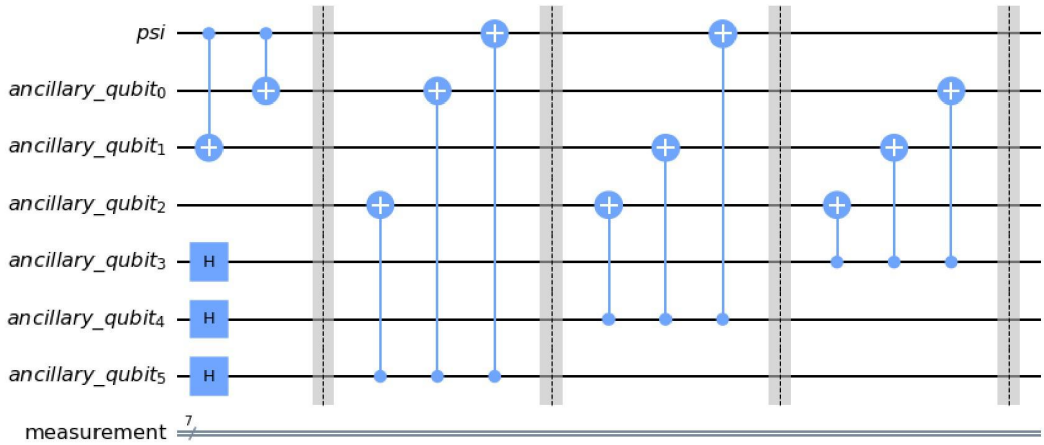


7-QUBIT Steane CODE, ENCODING CIRCUIT

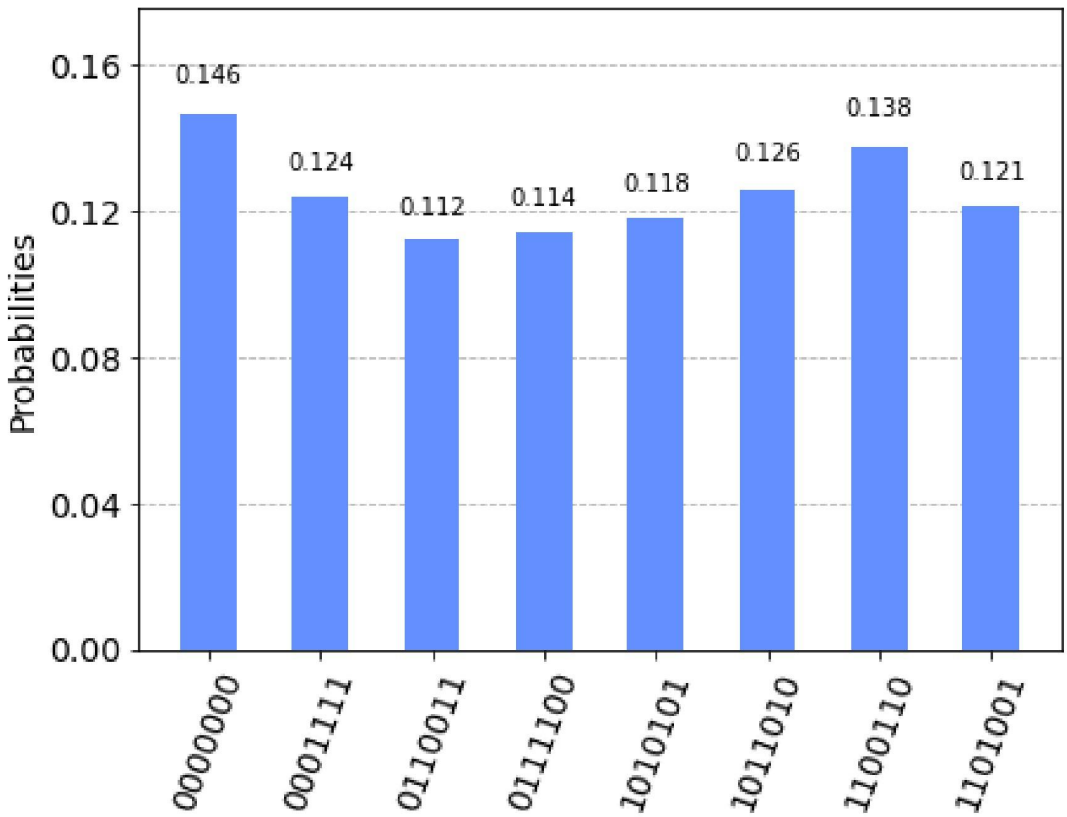
```
def seven_qubit_encode(qc):
    qc.h(6)
    qc.h(5)
    qc.h(4)
    qc.cx(0,2)
    qc.cx(0,1)
    qc.barrier()
    qc.cx(6,3)
    qc.cx(6,1)
    qc.cx(6,0)
    qc.barrier()
    qc.cx(5,3)
    qc.cx(5,2)
    qc.cx(5,0)
    qc.barrier()
    qc.cx(4,3)
    qc.cx(4,2)
    qc.cx(4,1)
    qc.barrier()
    return qc

aq = QuantumRegister(6,'ancillary_qubit')
init = QuantumRegister(1,'psi')
cb = ClassicalRegister(7,'measurement')
qc = QuantumCircuit(init,aq,cb)
```

```
qc = seven_qubit_encode(qc)
qc.draw(output='mpl')
```




```
qc.measure([0,1,2,3,4,5,6],[6,5,4,3,2,1,0])
qobj = assemble(qc)  # Assemble circuit into a Qobj that can be run
counts = sim.run(qobj).result().get_counts()  # Do the simulation,
returning the state vector
plot_histogram(counts)  # Display the output on measurement of state
vector
```

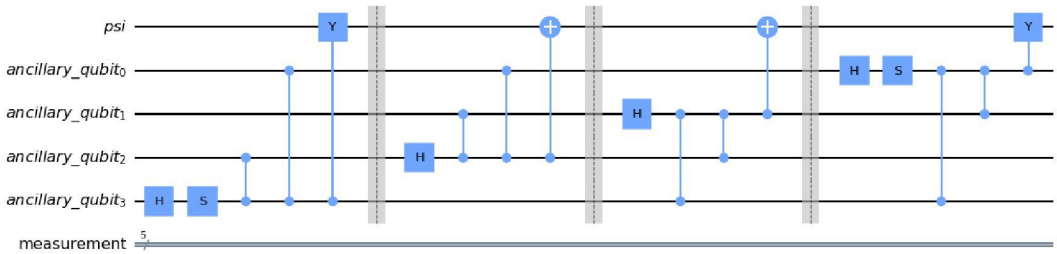


5-QUBIT CODE ENCODING CIRCUIT

```
#function from:
https://quantumcomputing.stackexchange.com/questions/14264/nielsenchuang-5-qubit-quantum-error-correction-encoding-gate
def five_encode(circ):

    circ.h(4)
    circ.s(4)
    # g1
    circ.cz(4,3)
    circ.cz(4,1)
    circ.cy(4,0)
    circ.barrier()
    circ.h(3)
    #g2
    circ.cz(3,2)
    circ.cz(3,1)
    circ.cx(3,0)
    circ.barrier()
    circ.h(2)
    #g3
    circ.cz(2,4)
    circ.cz(2,3)
    circ.cx(2,0)
    circ.barrier()
    circ.h(1)
    circ.s(1)
    #g4
    circ.cz(1,4)
    circ.cz(1,2)
    circ.cy(1,0)
    return circ

aq = QuantumRegister(4,'ancillary_qubit')
init = QuantumRegister(1,'psi')
cb = ClassicalRegister(5,'measurement')
qc1 = QuantumCircuit(init,aq,cb)
qc1 = five_encode(qc1)
qc1.draw(output='mpl')
```




```
qc1.measure([0,1,2,3,4],[4,3,2,1,0])
qobj = assemble(qc1)  # Assemble circuit into a Qobj that can be run
counts = sim.run(qobj).result().get_counts()  # Do the simulation,
returning the state vector
plot_histogram(counts)  # Display the output on measurement of state
vector
```