Title: Subsystem Quantum Error Correcting Codes

Date: Jun 01, 2007 11:40 AM

URL: http://pirsa.org/07060029

Abstract: <span>The essential insight of quantum error correction was that quantum information can be protected by suitably encoding this quantum information across multiple independently erred quantum systems. Recently it was realized that, since the most general method for encoding quantum information is to encode it into a subsystem, there exists a novel form of quantum error correction beyond the traditional quantum error correcting subspace codes. These new quantum error correcting subsystem codes differ from subspace codes in that their quantum correcting routines can be considerably simpler than related subspace codes. Here we present a class of quantum error correcting subsystem codes constructed from two classical linear codes. These codes are the subsystem versions of the quantum error correcting subspace codes which are generalizations of Shorâ€™s original quantum error correcting subspace codes. For every Shor-type code, the codes we present give a considerable savings in the number of stabilizer measurements needed in their error recovery routines.</span>

IV Canadian Quantum Information Student's Conference

Perimeter Institute

Waterloo, June 1-5 2007

Quantum Error Correction

Fault tolerant Quantum Computation

&

-Subsystem Codes-

Andrea Casaccino

Information Engineering Department

University of Siena

Italy

casaccino@dii.unisi.it

Dave Bacon

CSE Department

University of Washington

Seattle, USA

dabacon@cs.washington.edu

Talk Roadmap

- Quantum noise a

Talk Roadmap

- Quantum noise and environment interaction
- Classical Er

Talk Roadmap

- *Quantum noise and environment interaction*
- *Classical Error Correcting Codes and Quantum Error Correction*
- *Stabilizer Quantum Error Correction*
- *Subsystem Coding*
- *Bacon-Shor's Codes*

Talk Roadmap

- *Quantum noise and environment interaction*
- *Classical Error Correcting Codes and Quantum Error Correction*
- *Stabilizer Quantum Error Correction*
- *Subsystem Coding*
- *Bacon-Shor's Codes*

Quantum Error Correction

Quantum Computing Problems

- Schroedinger equation describes closed systems evolution.
- Real systems cannot be considered closed because of the environment interaction.
- The *Quantum operations formalism*, is a toolset to describe quantum noise and open systems behaviour.
- Quantum computers need a full control of the quantum interactions to be reliable.

Environment interact

Environment interaction

Open system analysis describes the environment and the system as a unique object.

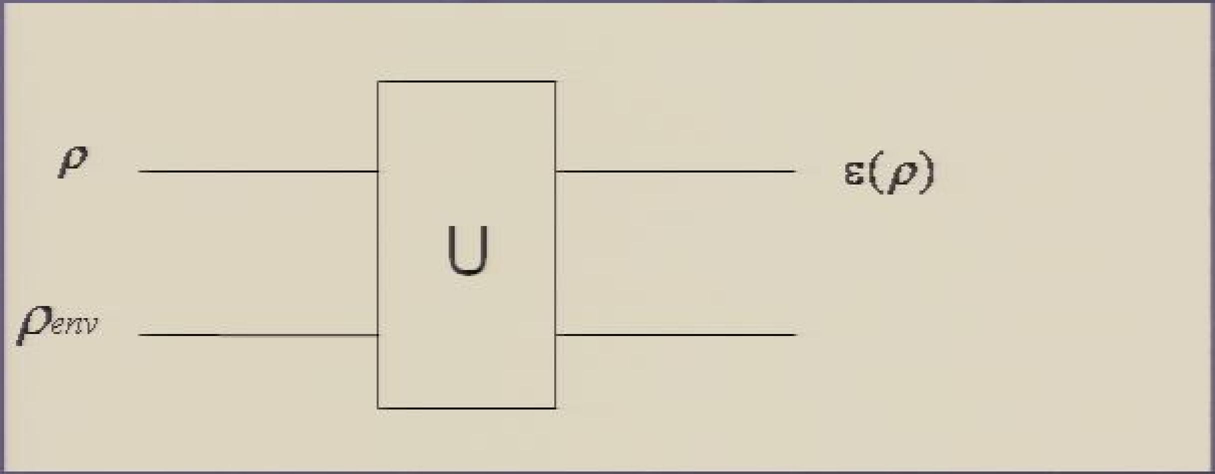
$$\rho' = \mathcal{E}(\rho).$$

Environment interaction

Open system analysis describes the environment and the system as a unique object.

Environment and the open system can be considered as a closed system

$$\rho' = \mathcal{E}(\rho).$$

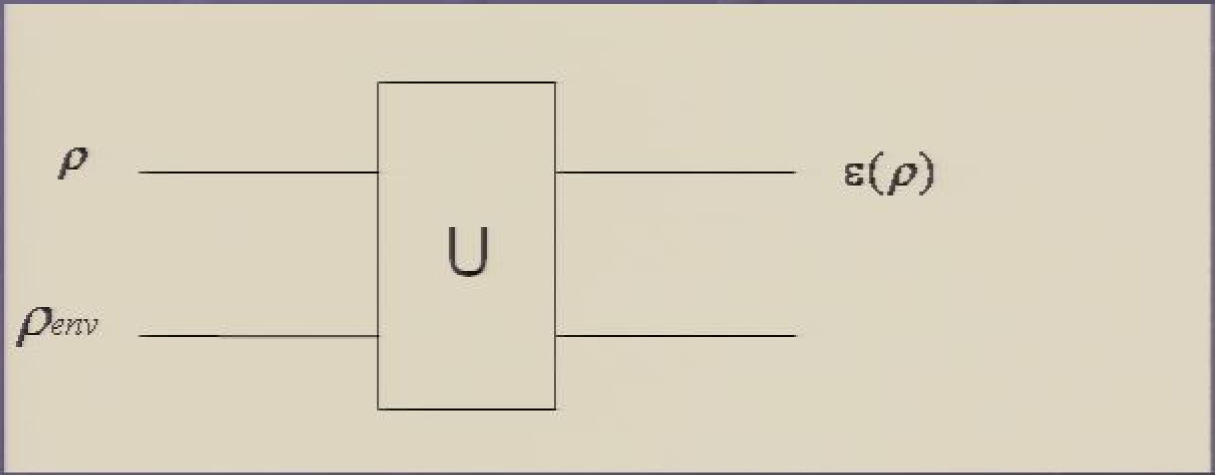


Environment interaction

Open system analysis describes the environment and the system as a unique object.

Environment and the open system can be considered as a closed system

$$\rho' = \mathcal{E}(\rho).$$



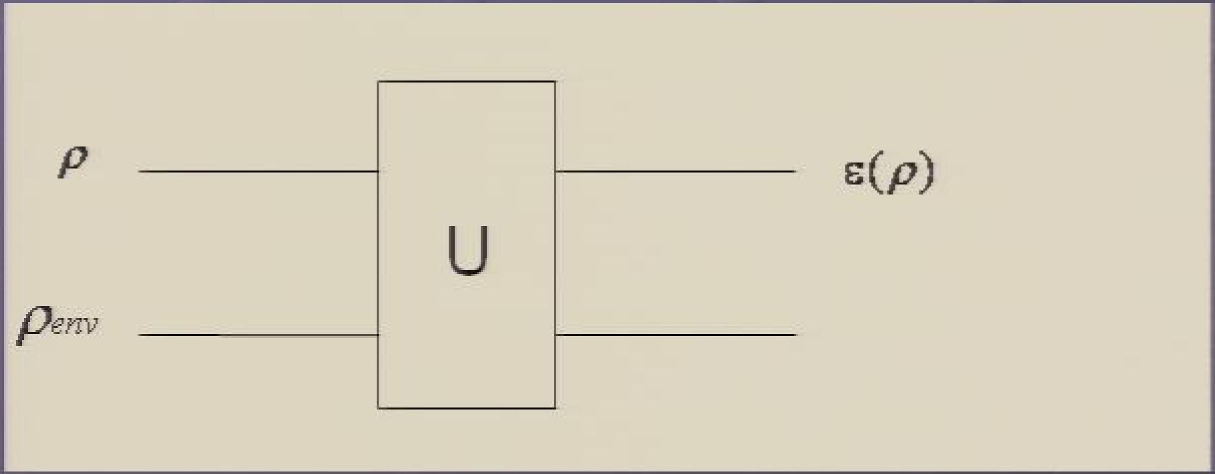
$$\mathcal{E}(\rho) = tr_{env}\left[U(\rho \otimes \rho_{env})U^{\dagger}\right]$$

Environment *interaction*

Open system analysis describes the environment and the system as a unique object.

Environment and the open system can be considered as a closed system

$$\rho' = \mathcal{E}(\rho).$$



$$\mathcal{E}(\rho) = tr_{env}\left[U(\rho \otimes \rho_{env})U^{\dagger}\right]$$

The partial trace operator gives back the state of the system after the interaction with the environment

Noise Linearization

- [illegible]

Noise Linearization

- Take an orthonormal basis for the environment $|e_k\rangle$ and $\rho_{env} = |e_0\rangle\langle e_0|$ the initial state. Then is possible to write the environment interaction as

$$\mathcal{E}(\rho) = \sum_k \langle e_k | U \left[(\rho \otimes |e_0\rangle\langle e_0| \right] U^\dagger | e_k \rangle = \sum_k E_k \rho E_k^\dagger$$

Noise Linearization

- Take an orthonormal basis for the environment $|e_k\rangle$ and $\rho_{env} = |e_0\rangle\langle e_0|$ the initial state. Then is possible to write the environment interaction as

$$\mathcal{E}(\rho) = \sum_k \langle e_k | U \left[(\rho \otimes |e_0\rangle\langle e_0|) \right] U^\dagger | e_k \rangle = \sum_k E_k \rho E_k^\dagger$$

$$\Downarrow$$

$$\mathcal{E}(\rho) = \sum_k E_k \rho E_k^\dagger$$

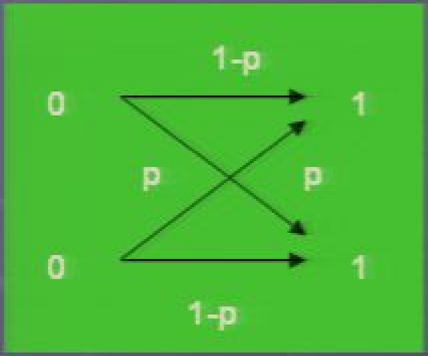
$0<k<d^2$

Noise ope

Noise operators on qubit

$$\varepsilon(\rho) = E_0 \rho E_0^\dagger + E_1 \rho E_1^\dagger$$

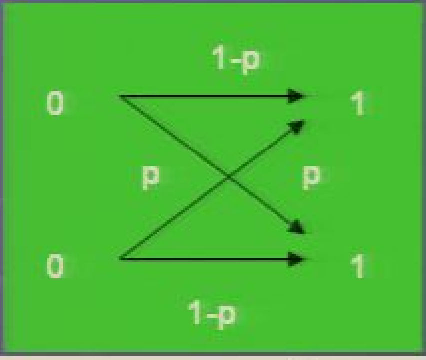
Channel Model



Noise operators on qubit

$$\varepsilon(\rho) = E_0 \rho E_0^\dagger + E_1 \rho E_1^\dagger$$

Channel Model



- Bit flip

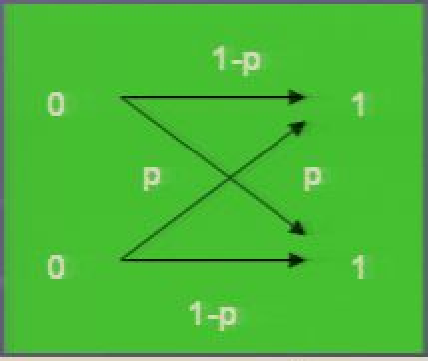
$$E_1 = \sqrt{1-p}\,X = \sqrt{1-p}\cdot\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

$$E_0 = \sqrt{p}\,I = \sqrt{p}\cdot\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

Noise operators on qubit

$$\varepsilon(\rho) = E_0 \rho E_0^\dagger + E_1 \rho E_1^\dagger$$

Channel Model



- Bit flip

$$E_1 = \sqrt{1-p}\,X = \sqrt{1-p} \cdot \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

$$E_0 = \sqrt{p}\,I = \sqrt{p} \cdot \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

- Phase flip

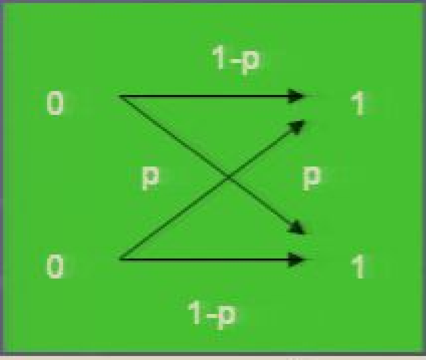
$$E_0 = \sqrt{p}I = \sqrt{p} \cdot \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

$$E_1 = \sqrt{1-p}Z = \sqrt{1-p} \cdot \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

Noise operators on qubit

$$\varepsilon(\rho) = E_0 \rho E_0^\dagger + E_1 \rho E_1^\dagger$$

Channel Model



- Bit flip

$$E_1 = \sqrt{1-p}\,X = \sqrt{1-p}\cdot\begin{pmatrix}0 & 1\\ 1 & 0\end{pmatrix}$$

$$E_0 = \sqrt{p}\,I = \sqrt{p}\cdot\begin{pmatrix}1 & 0\\ 0 & 1\end{pmatrix}$$

- Phase flip

$$E_0 = \sqrt{p}I = \sqrt{p}\cdot\begin{pmatrix}1 & 0\\ 0 & 1\end{pmatrix}$$

$$E_1 = \sqrt{1-p}Z = \sqrt{1-p}\cdot\begin{pmatrix}1 & 0\\ 0 & -1\end{pmatrix}$$

- Bit- Phase flip

$$E_0 = \sqrt{p}I = \sqrt{p}\cdot\begin{pmatrix}1 & 0\\ 0 & 1\end{pmatrix}$$

$$E_1 = \sqrt{1-p}Y = \sqrt{1-p}\cdot\begin{pmatrix}0 & -i\\ i & 0\end{pmatrix}$$

Classical Error correction

- The key idea of error correction is that we need to add redundancy bits to protect Information from noise. In this way it is possible (under certain conditions) to rebuild the information content.
- Linear error correcting codes are the simplest ones.
- A linear code is defined as a vector subspace on ***GF(n)*** .

[illegible]

Classical Error correction

- The key idea of error correction is that we need to add redundancy bits to protect Information from noise. In this way it is possible (under certain conditions) to rebuild the information content.
- Linear error correcting codes are the simplest ones.
- A linear code is defined as a **vector subspace** on ***GF(n)*** .
- E.G.: *ASCII code* is a linear code on $GF(2)$.

Linear Error correcting c

Linear Error correcting codes

- An n bit information coding with a k bit CODE is defined by a generator matrix G nxk whose elements belong to $GF(2)$.
- An x message *of n bits*, is encoded in a message x_c of k bit by the following operation

$$\boxed{x_c = Gx}$$

Linear Error correcting codes

- An *n* bit information coding with a k bit CODE is defined by a generator matrix *G nxk* whose elements belong to *GF(2)*.
- An x message *of n bits,* is encoded in a message x_c of k bit by the following operation

$$\boxed{x_c = Gx}$$

a k bit coding used to encode n bit of Information is denoted by the [n,k] notation

Classical error co

Classical error correcting Codes(2)

- To define classical linear codes it is often used the Parity check matrix.
- In this way a [n,k] encoding is defined as the complete vector set x of n elements on *GF(2)* with the property:

$$Hx = 0$$

Classical error correcting Codes(2)

- To define classical linear codes it is often used the Parity check matrix.
- In this way a [n,k] encoding is defined as the complete vector set x of n elements on $GF(2)$ with the property:

$$\mathbf{H}x=0$$

where **H** is a $n\text{-}k \times n$ matrix .

- Moreover suppose that x and y are two n bit words: the **Hamming distance** between x and y is defined as the number of bits (in the same position) the two words differ from each other. For example:

$$d((1,1,0,0),(0,1,0,1))=2$$

Classical error correcting Codes(2)

- To define classical linear codes it is often used the Parity check matrix.
- In this way a [n,k] encoding is defined as the complete vector set x of n elements on *GF(2)* with the property:

$$\mathrm{H}x=0$$

where **H** is a $n\text{-}k \times n$ matrix .

- Moreover suppose that x and y are two n bit words: the **Hamming distance** between x and y is defined as the number of bits (in the same position) the two words differ from each other. For example:

$$d((1,1,0,0),(0,1,0,1))=2$$

- In particoular the Code Hamming distance is defined as :

$$d(C) \equiv \min_{x,y\in C,x\neq y} d(x,y)$$

An example:

Th

An example:
The Repetition Code

An example:
The Repetition Code

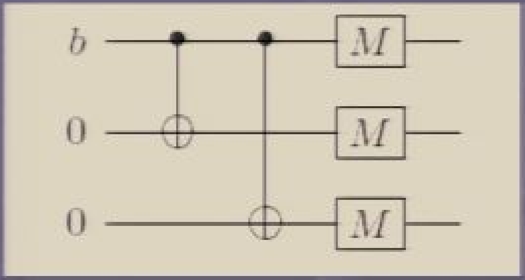
- The simplest way to protect one bit is to copy it three times…

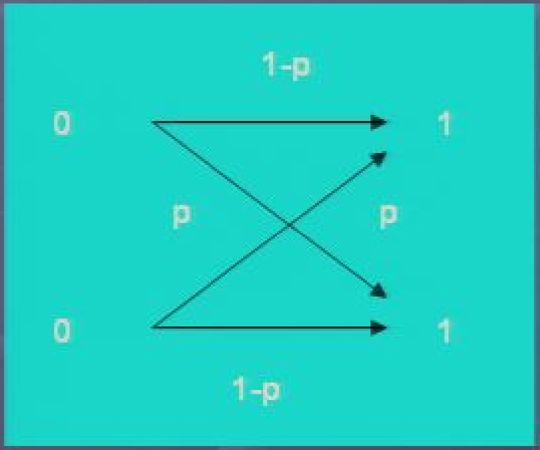
$$0 \rightarrow 000$$
$$1 \rightarrow 111$$

An example:
The Repetition Code

- The simplest way to protect one bit is to copy it three times…

$$0 \rightarrow 000$$
$$1 \rightarrow 111$$





$$M = \begin{bmatrix} 1-p & p \\ p & 1-p \end{bmatrix}$$

Suppose to have a 001 string for the channel output, the decoding circuit, "chooses" that the most likely event has been the third bit flip and it will decode the output as a 0.

Quantum codes

- We have to deal with three major problems when we want to build a quantum code :
  - ***No cloning theorem*** : it's impossible to duplicate an unknow quantum state.
  - ***Quantum errors is a continous set :*** it *seems* to be a non trivial operation to recognize which kind of error has occurred.
  - [illegible]

Quantum codes

- We have to deal with three major problems when we want to build a quantum code :
  - ***No cloning theorem*** : it's impossible to duplicate an unknow quantum state.
  - ***Quantum errors is a continous set :*** it *seems* to be a non trivial operation to recognize which kind of error has occurred.
  - ***Measuring destroys quantum Information***: we have to extract the output of the channel with a measure to know if the message transmitted is correct and thus correctly decoded .

Quantum codes

- We have to deal with three major problems when we want to build a quantum code :
  - ***No cloning theorem*** : it's impossible to duplicate an unknow quantum state.
  - ***Quantum errors is a continous set :*** it *seems* to be a non trivial operation to recognize which kind of error has occurred.
  - ***Measuring destroys quantum Information***: we have to extract the output of the channel with a measure to know if the message transmitted is correct and thus correctly decoded .

Quantum Error Correction

- Quantum error correcting rules are similar to classical ones and are denoted by the *[[n,k,d]]* notation where n is the number of qubit of the encoding, k is the number of encoded qubit while d is the code distance.

Quantum Error Correction

- Quantum error correcting rules are similar to classical ones and are denoted by the *[[n,k,d]]* notation where n is the number of qubit of the encoding, k is the number of encoded qubit while d is the code distance.

- We will make the assumption that quantum noise is described by a quantum operation $\mathcal{E}$ and error recovery is made by a quantum operation $\mathfrak{R}$

Quantum Error Correction

- Quantum error correcting rules are similar to classical ones and are denoted by the *[[n,k,d]]* notation where n is the number of qubit of the encoding, k is the number of encoded qubit while d is the code distance.

- We will make the assumption that quantum noise is described by a quantum operation $\mathcal{E}$ and error recovery is made by a quantum operation $\Re$

- With this hypothesis we require that for every quantum state ρ the following equation holds

$$(\Re \circ \mathcal{E})(\rho) = \rho$$

Quantum Error Correction

- Quantum error correcting rules are similar to classical ones and are denoted by the *[[n,k,d]]* notation where n is the number of qubit of the encoding, k is the number of encoded qubit while d is the code distance.

- We will make the assumption that quantum noise is described by a quantum operation $\mathcal{E}$ and error recovery is made by a quantum operation $\Re$

- With this hypothesis we require that for every quantum state ρ the following equation holds

$$(\Re \circ \mathcal{E})(\rho) = \rho$$

Quantum error correction cond

Quantum error correction conditions

and errors discret

Quantum error correction conditions and errors discretization

- Let $|\phi_i\rangle$ be a Code orthonormal basis,such that is the code space.

Quantum error correction conditions and errors discretization

- Let $|\phi_i\rangle$ be a Code orthonormal basis,such that $H_C = span\{|\phi_i\rangle\}$ is the code space.

Quantum error correction conditions and errors discretization

- Let $|\phi_i\rangle$ be a Code orthonormal basis,such that $H_C = span\{|\phi_i\rangle\}$ is the code space.
- Suppose $\mathcal{E}$ a quantum operation with operation elements $\{E_i\}$. Then a necessary and sufficient condition for the existence of an error-correction operation $\Re$ correcting $\mathcal{E}$ on C is that

$$\langle \phi_i | E_i^\dagger E_j | \phi_j \rangle = C_{ij}\delta_{ij}$$

In this case $\{E_i\}$ is said Correctable error Set for the quantum operation

Quantum error correction conditions and errors discretization

- Let $|\phi_i\rangle$ be a Code orthonormal basis,such that $H_C = span\{|\phi_i\rangle\}$ is the code space.
- Suppose $\mathcal{E}$ a quantum operation with operation elements $\{E_i\}$. Then a necessary and sufficient condition for the existence of an error-correction operation $\mathfrak{R}$ correcting $\mathcal{E}$ on C is that

$$\langle \phi_i | E_i^{\dagger} E_j | \phi_j \rangle = C_{ij}\delta_{ij}$$

In this case $\{E_i\}$ is said Correctable error Set for the quantum operation

- In addition for a [[n,k,d]] code is true the following equation

$$d=2t+1$$

Quantum error correction conditions and errors discretization

- Let $|\phi_i\rangle$ be a Code orthonormal basis,such that $H_C = span\{|\phi_i\rangle\}$ is the code space.
- Suppose $\mathcal{E}$ a quantum operation with operation elements $\{E_i\}$. Then a necessary and sufficient condition for the existence of an error-correction operation $\mathfrak{R}$ correcting $\mathcal{E}$ on C is that

$$\langle\phi_i|E_i^\dagger E_j|\phi_j\rangle = C_{ij}\delta_{ij}$$

In this case $\{E_i\}$ is said Correctable error Set for the quantum operation

- In addition for a [[n,k,d]] code is true the following equation

$$d=2t+1$$

where t is the number of correctable errors

Three qubit code

We encode the single qubit s

Three qubit code

We encode the single qubit state

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

Three qubit code

We encode the single qubit state

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

With the following encoding

$$|0\rangle \rightarrow |0_L\rangle \equiv |000\rangle$$
$$|1\rangle \rightarrow |1_L\rangle \equiv |111\rangle$$

Three qubit code

We encode the single qubit state

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

With the following encoding

$$|0\rangle \rightarrow |0_L\rangle \equiv |000\rangle$$
$$|1\rangle \rightarrow |1_L\rangle \equiv |111\rangle$$

Where the quantum superposition is taken in the encoded qubit superpo

Three qubit code

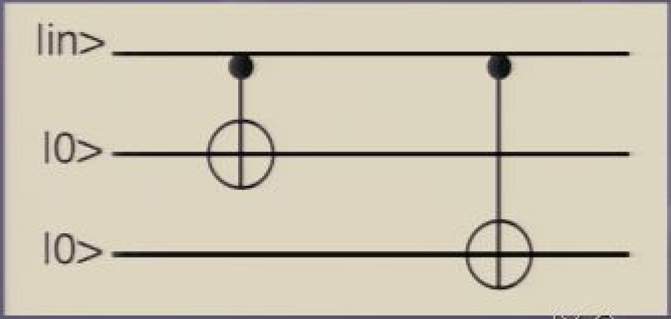
We encode the single qubit state

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

With the following encoding

$$|0\rangle \rightarrow |0_L\rangle \equiv |000\rangle$$
$$|1\rangle \rightarrow |1_L\rangle \equiv |111\rangle$$

Where the quantum superposition is taken in the encoded qubit superposition



$$\$ = \begin{cases} E_0 = \sqrt{1-p}I, \\ E_1 = \sqrt{p}X \end{cases}$$

Three qubit code

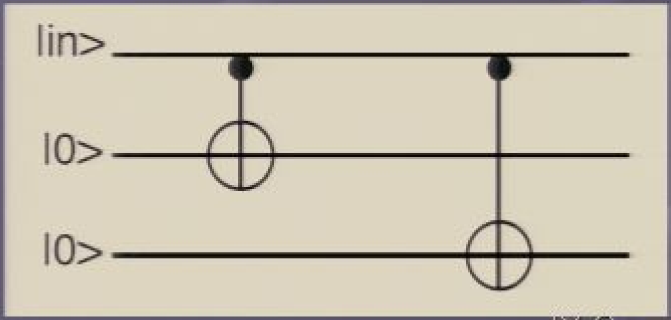
We encode the single qubit state

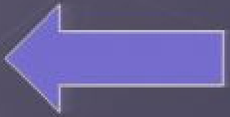
$$|\psi\rangle = a|0\rangle + b|1\rangle \rightarrow |\psi\rangle = a|000\rangle + b|111\rangle$$

With the following encoding

$$|0\rangle \rightarrow |0_L\rangle \equiv |000\rangle$$
$$|1\rangle \rightarrow |1_L\rangle \equiv |111\rangle$$

Where the quantum superposition is taken in the encoded qubit superposition





$$\$ = \begin{cases} E_0 = \sqrt{1-p}I, \\ E_1 = \sqrt{p}X \end{cases}$$

$\mathcal{E}$

Error detection

- Errors are detected by a decoding-recovery circuit .
- The error syndromes are obtained by 4 projective measurements

$P_0 \equiv |000\rangle\langle 000| + |111\rangle\langle 111|$ no errors

Error detection

- Errors are detected by a decoding-recovery circuit .
- The error syndromes are obtained by 4 projective measurements

$P_0 \equiv |000\rangle\langle 000| + |111\rangle\langle 111|$ no errors

$P_1 \equiv |100\rangle\langle 100| + |011\rangle\langle 011|$ First qubit bit flip

Error detection

- Errors are detected by a decoding-recovery circuit .
- The error syndromes are obtained by 4 projective measurements

$P_0 \equiv |000\rangle\langle 000| + |111\rangle\langle 111|$ no errors

$P_1 \equiv |100\rangle\langle 100| + |011\rangle\langle 011|$ First qubit bit flip

$P_2 \equiv |010\rangle\langle 010| + |101\rangle\langle 101|$ Second qubit bit flip

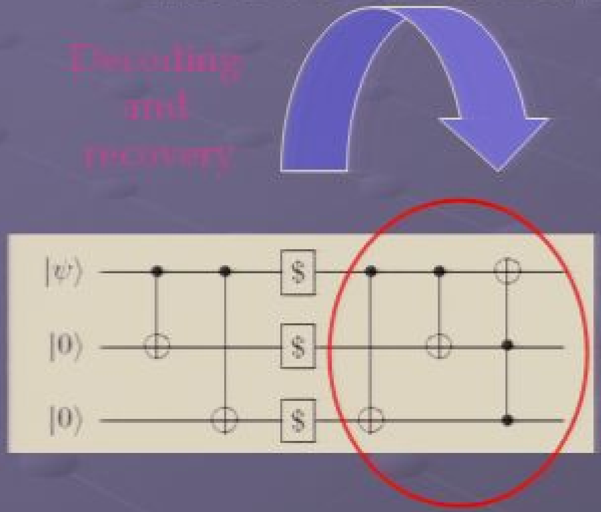
Error detection

- Errors are detected by a decoding-recovery circuit .
- The error syndromes are obtained by 4 projective measurements

$P_0 \equiv \|000\rangle\langle 000\| + \|111\rangle\langle 111\|$	no errors
$P_1 \equiv \|100\rangle\langle 100\| + \|011\rangle\langle 011\|$	First qubit bit flip
$P_2 \equiv \|010\rangle\langle 010\| + \|101\rangle\langle 101\|$	Second qubit bit flip
$P_3 \equiv \|001\rangle\langle 001\| + \|110\rangle\langle 110\|$	Third qubit bit flip

Error detection

- Errors are detected by a decoding-recovery circuit .
- The error syndromes are obtained by 4 projective measurements



$P_0 \equiv \|000\rangle\langle 000\| + \|111\rangle\langle 111\|$	no errors
$P_1 \equiv \|100\rangle\langle 100\| + \|011\rangle\langle 011\|$	First qubit bit flip
$P_2 \equiv \|010\rangle\langle 010\| + \|101\rangle\langle 101\|$	Second qubit bit flip
$P_3 \equiv \|001\rangle\langle 001\| + \|110\rangle\langle 110\|$	Third qubit bit flip

Pauli Group and Stabilizer Codes

- Pauli matrixes form an algebric group called Pauli group, G_n with the tensor product of n Pauli operators each of them acting on one of the n qubit. For example

$$Z_1 Z_2 = Z \otimes Z \otimes I \qquad Z_2 Z_3 = I \otimes Z \otimes Z$$

Pauli Group and Stabilizer Codes

- Pauli matrixes form an algebric group called Pauli group, G_n with the tensor product of n Pauli operators each of them acting on one of the n qubit. For example

$$Z_1Z_2 = Z \otimes Z \otimes I \qquad Z_2Z_3 = I \otimes Z \otimes Z$$

- The Stabilizer subgroup is defined as the set of Pauli operators that "stabilize" the code subspace (+1 eigenvalue). For example for the repetition code subspace we have

$$|000\rangle, |111\rangle \longleftrightarrow S = \langle Z_1Z_2, Z_2Z_3 \rangle$$

Error detection tools

- Stabilizer

Error detection tools

- Stabilizer Codes help to simplify error detection procedu

Error detection tools

- Stabilizer Codes help to simplify error detection procedures.
- In fact the +1,-1 eigenvalues of the generat

Error detection tools

- Stabilizer Codes help to simplify error detection procedures.
- In fact the +1,-1 eigenvalues of the generators measurings on the state and the commutati

Error detection tools

- Stabilizer Codes help to simplify error detection procedures.
- In fact the +1,-1 eigenvalues of the generators measurings on the state and the commutation rules of Pauli group operators are the ingred

Error detection tools

- Stabilizer Codes help to simplify error detection procedures.
- In fact the +1,-1 eigenvalues of the generators measurings on the state and the commutation rules of Pauli group operators are the ingredients to detect any kind of errors

$$X_1|\psi\rangle = |\varphi\rangle$$
$$Z_1Z_2|\varphi\rangle = Z_1Z_2\left(X_1|\psi\rangle\right) =$$
$$= -X_1Z_1Z_2|\psi\rangle = -X_1|\psi\rangle = -|\varphi\rangle$$
$$-Z_1Z_2|\varphi\rangle = |\varphi\rangle$$

Error detection tools

- Stabilizer Codes help to simplify error detection procedures.
- In fact the +1,-1 eigenvalues of the generators measurings on the state and the commutation rules of Pauli group operators are the ingredients to detect any kind of errors

$$X_1|\psi\rangle = |\varphi\rangle$$
$$Z_1Z_2|\varphi\rangle = Z_1Z_2\left(X_1|\psi\rangle\right) =$$
$$= -X_1Z_1Z_2|\psi\rangle = -X_1|\psi\rangle = -|\varphi\rangle$$
$$-Z_1Z_2|\varphi\rangle = |\varphi\rangle$$

$$\left[\bar{X}_i, \bar{X}_i\right] = 0$$
$$\left[\bar{Z}_i, \bar{Z}_i\right] = 0$$
$$\left\{\bar{X}_i, \bar{Z}_i\right\} = 0$$
$$\left\{\bar{X}_i, \bar{Y}_i\right\} = 0$$
$$\left\{\bar{Z}_i, \bar{Y}_i\right\} = 0.$$

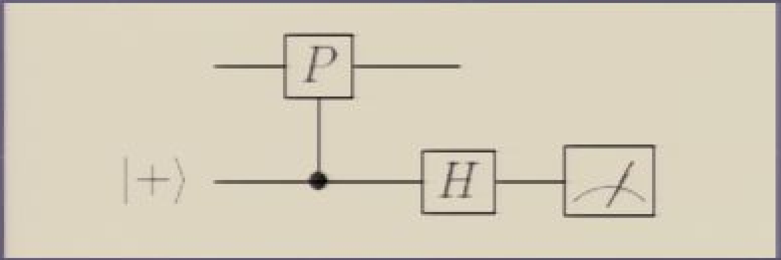
Commutation rules

Error detection tools

- Stabilizer Codes help to simplify error detection procedures.
- In fact the +1,-1 eigenvalues of the generators measurings on the state and the commutation rules of Pauli group operators are the ingredients to detect any kind of errors

$$X_1|\psi\rangle = |\varphi\rangle$$
$$Z_1Z_2|\varphi\rangle = Z_1Z_2\left(X_1|\psi\rangle\right) =$$
$$= -X_1Z_1Z_2|\psi\rangle = -X_1|\psi\rangle = -|\varphi\rangle$$
$$-Z_1Z_2|\varphi\rangle = |\varphi\rangle$$

$$\left[\bar{X}_i, \bar{X}_i\right] = 0$$
$$\left[\bar{Z}_i, \bar{Z}_i\right] = 0$$
$$\left\{\bar{X}_i, \bar{Z}_i\right\} = 0$$
$$\left\{\bar{X}_i, \bar{Y}_i\right\} = 0$$
$$\left\{\bar{Z}_i, \bar{Y}_i\right\} = 0.$$



Error detect

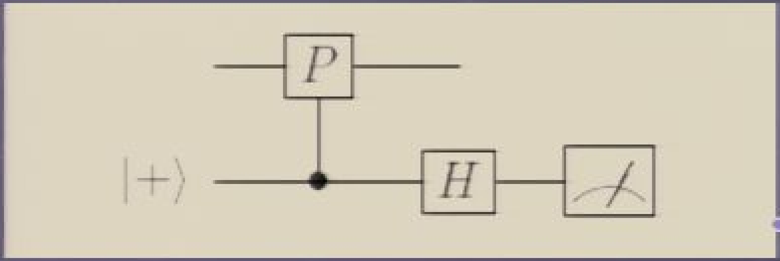
Commutation rules

Error detection tools

- Stabilizer Codes help to simplify error detection procedures.
- In fact the +1,-1 eigenvalues of the generators measurings on the state and the commutation rules of Pauli group operators are the ingredients to detect any kind of errors

$$X_1|\psi\rangle = |\varphi\rangle$$
$$Z_1Z_2|\varphi\rangle = Z_1Z_2\left(X_1|\psi\rangle\right) =$$
$$= -X_1Z_1Z_2|\psi\rangle = -X_1|\psi\rangle = -|\varphi\rangle$$
$$-Z_1Z_2|\varphi\rangle = |\varphi\rangle$$

$$\left[\bar{X}_i, \bar{X}_i\right] = 0$$
$$\left[\bar{Z}_i, \bar{Z}_i\right] = 0$$
$$\left\{\bar{X}_i, \bar{Z}_i\right\} = 0$$
$$\left\{\bar{X}_i, \bar{Y}_i\right\} = 0$$
$$\left\{\bar{Z}_i, \bar{Y}_i\right\} = 0.$$



Error detection circuit

Commutation rules

Example: Redundancy Code Error Syndromes

Z_1Z_2	Z_2Z_3	Errore detected	Recovery action
+1	+1	No errors	no
+1	-1	3° Bit flip	*Operator: IIX*
-1	+1	1° Bit flip	*Operator: XII*
-1	-1	2° Bit flip	*Operator: IXI*

Examples : Steane Code

Generator	Operators
S_1	$I\otimes X\otimes X\otimes I\otimes I\otimes X\otimes X$
S_2	$X\otimes I\otimes X\otimes I\otimes X\otimes I\otimes X$
S_3	$I\otimes I\otimes I\otimes X\otimes X\otimes X\otimes X$
S_4	$Z\otimes I\otimes Z\otimes I\otimes Z\otimes I\otimes Z$
S_5	$I\otimes I\otimes I\otimes Z\otimes Z\otimes Z\otimes Z$
S_6	$I\otimes Z\otimes Z\otimes I\otimes I\otimes Z\otimes Z$
$\bar{X}$	$X\otimes X\otimes X\otimes X\otimes X\otimes X\otimes X$
$\bar{Z}$	$Z\otimes Z\otimes Z\otimes Z\otimes Z\otimes Z\otimes Z$

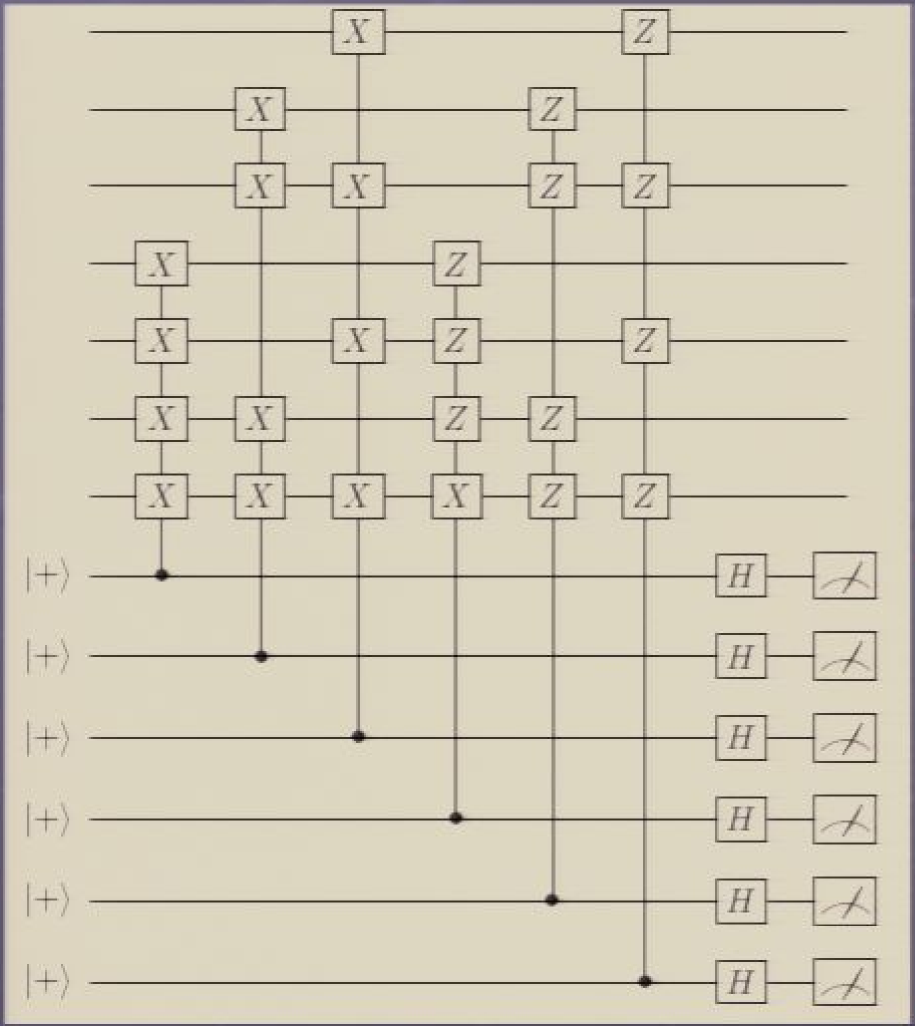
Logical operations

Operators that commute With the stabilizer subgroup: Error of this form can't be detected

Steane code encoding-decoding circuit

Stabilizer encoding

Ancilla qubits



Syndrome detection

Shor Code

$$|0_L\rangle = \frac{1}{2\sqrt{2}}(|000\rangle+|111\rangle)\otimes(|000\rangle+|111\rangle)\otimes(|000\rangle+|111\rangle)$$

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

Stabilizer generators:

$ZZIIIIIII$	$IIIZZIIII$	$IIIIIIZZI$	$XXXXXXIII$
$IZZIIIIII$	$IIIIZZIII$	$IIIIIIIZZ$	$IIIXXXXXX$

Define subspace:

$$S|\psi\rangle = |\psi\rangle$$

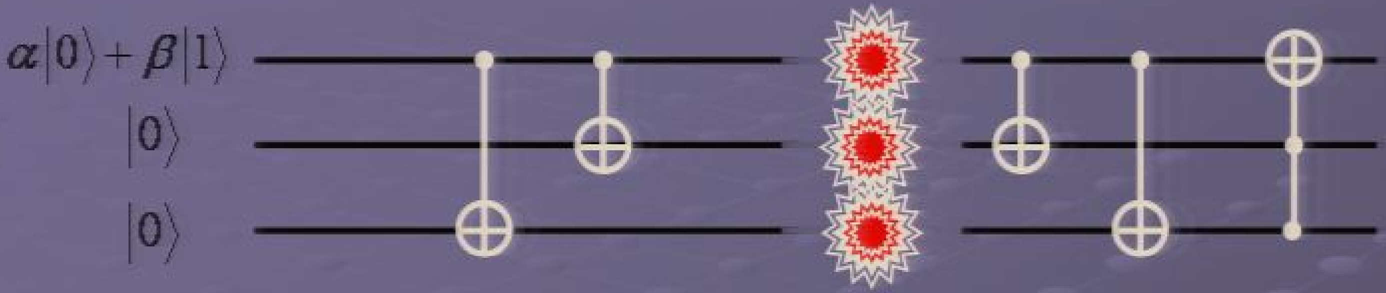
Logical operators:

$$\bar{Z} = ZZZZZZZZZ$$
$$\bar{X} = XXXXXXXXX$$

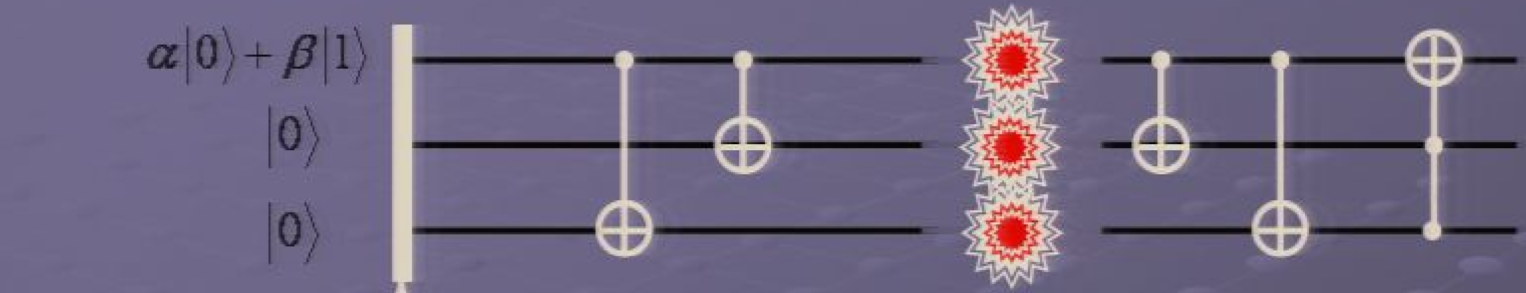
$$\bar{Z}|0_L\rangle = |0\rangle$$
$$\bar{Z}|1_L\rangle = -|1\rangle$$

Bit flip Cir

Bit flip Circuit

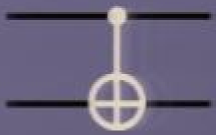


Bit flip Circuit

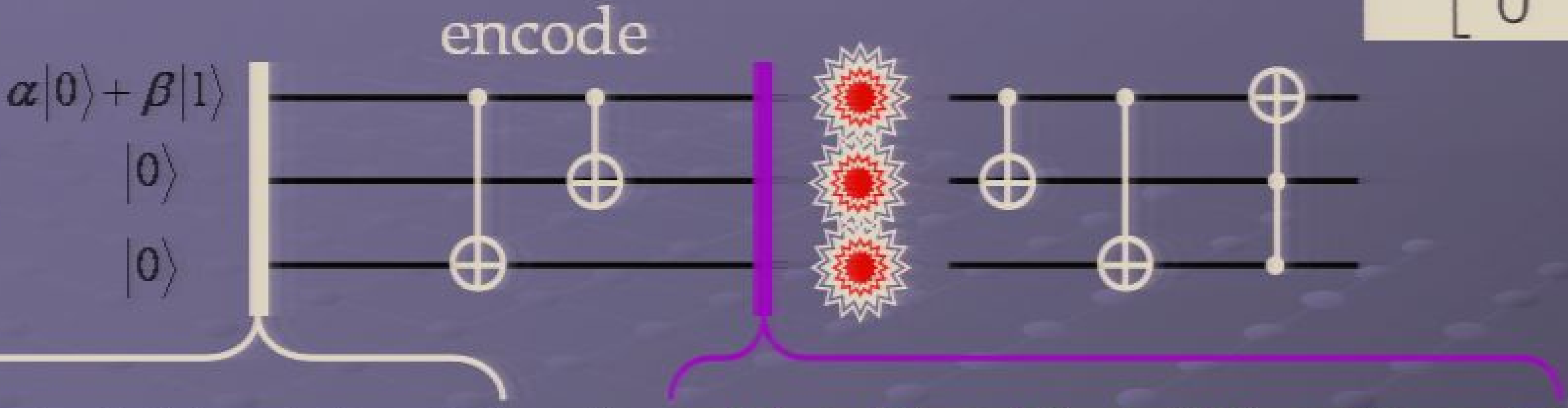


$$(\alpha|0\rangle + \beta|1\rangle) \otimes |0\rangle \otimes |0\rangle$$

Bit flip Circuit

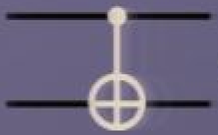


$$= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

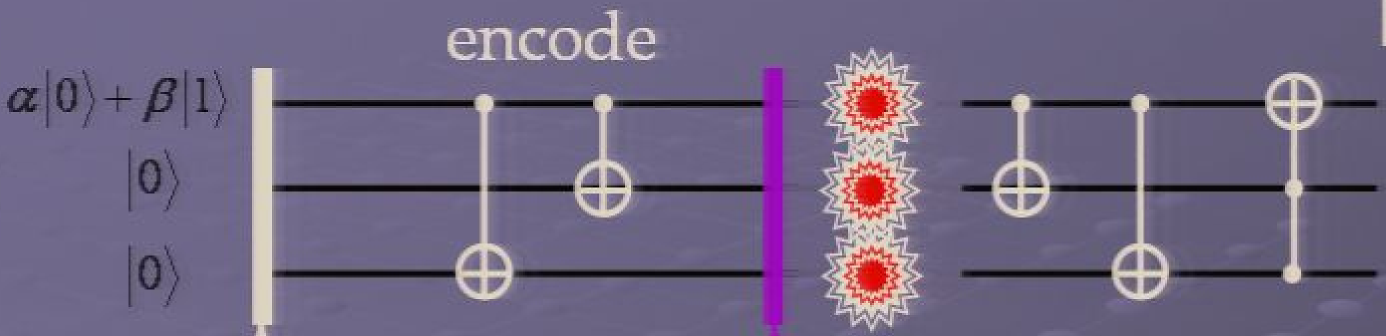


$$(\alpha|0\rangle + \beta|1\rangle) \otimes |0\rangle \otimes |0\rangle \xrightarrow{U} \alpha|0\rangle \otimes |0\rangle \otimes |0\rangle + \beta|1\rangle \otimes |1\rangle \otimes |1\rangle$$

Bit flip Circuit



$$= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$



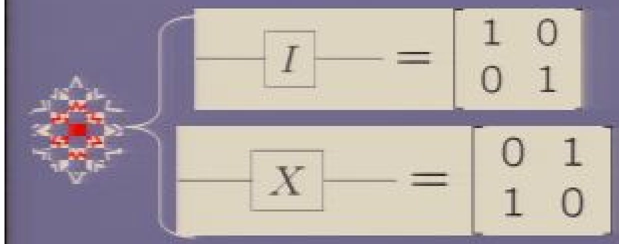
$$(\alpha|0\rangle+\beta|1\rangle)\otimes|0\rangle\otimes|0\rangle \xrightarrow{U} \alpha|0\rangle\otimes|0\rangle\otimes|0\rangle+\beta|1\rangle\otimes|1\rangle\otimes|1\rangle$$

$$\alpha|000\rangle+\beta|111\rangle \overset{\text{XII}}{\rightsquigarrow} \alpha|100\rangle+\beta|011\rangle \to (\alpha|1\rangle+\beta|0\rangle)|11\rangle \to (\alpha|0\rangle+\beta|1\rangle)|11\rangle$$

$$\alpha|000\rangle+\beta|111\rangle \overset{\text{IXI}}{\rightsquigarrow} \alpha|010\rangle+\beta|101\rangle \to (\alpha|0\rangle+\beta|1\rangle)|10\rangle \to (\alpha|0\rangle+\beta|1\rangle)|10\rangle$$

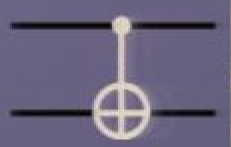
$$\alpha|000\rangle+\beta|111\rangle \overset{\text{IIX}}{\rightsquigarrow} \alpha|001\rangle+\beta|110\rangle \to (\alpha|0\rangle+\beta|1\rangle)|01\rangle \to (\alpha|0\rangle+\beta|1\rangle)|01\rangle$$

Bit flip Circuit

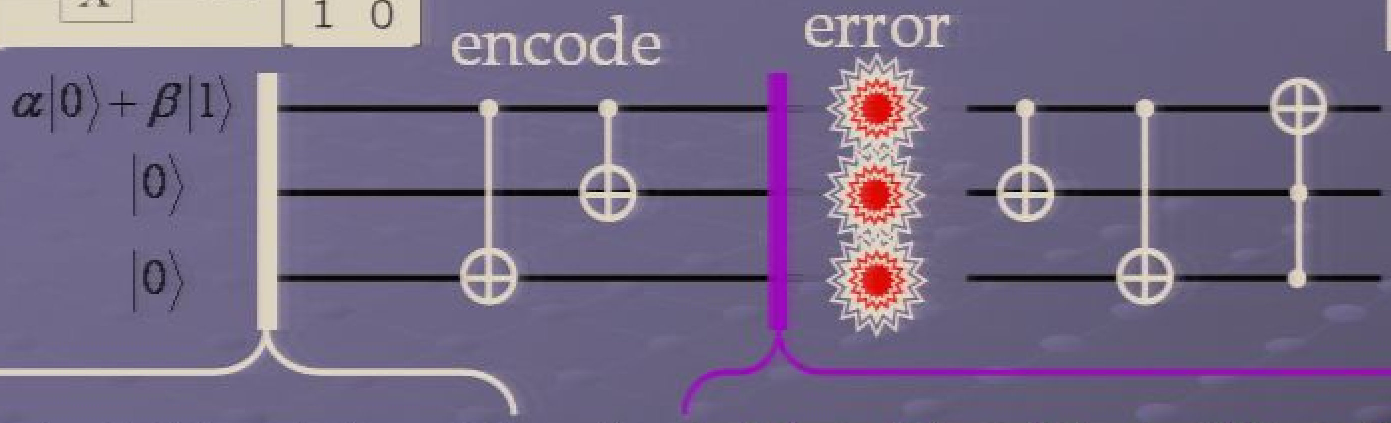


$$I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$



$$= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$



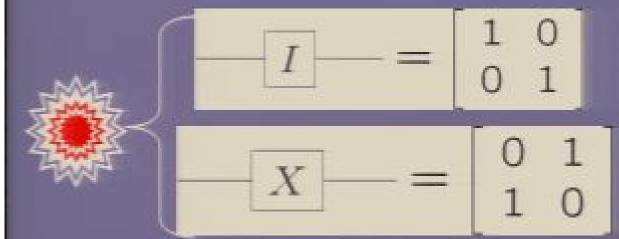
$$(\alpha|0\rangle + \beta|1\rangle) \otimes |0\rangle \otimes |0\rangle \xrightarrow{U} \alpha|0\rangle \otimes |0\rangle \otimes |0\rangle + \beta|1\rangle \otimes |1\rangle \otimes |1\rangle$$

$$\alpha|000\rangle + \beta|111\rangle \overset{XII}{\rightarrow} \alpha|100\rangle + \beta|011\rangle \rightarrow (\alpha|1\rangle + \beta|0\rangle)\,|11\rangle \rightarrow (\alpha|0\rangle + \beta|1\rangle)\,|11\rangle$$

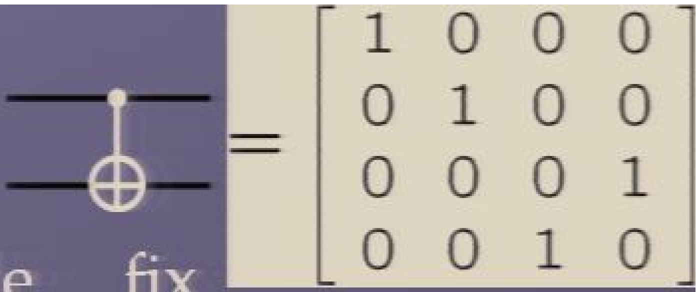
$$\alpha|000\rangle + \beta|111\rangle \overset{IXI}{\rightarrow} \alpha|010\rangle + \beta|101\rangle \rightarrow (\alpha|0\rangle + \beta|1\rangle)\,|10\rangle \rightarrow (\alpha|0\rangle + \beta|1\rangle)\,|10\rangle$$

$$\alpha|000\rangle + \beta|111\rangle \overset{IIX}{\rightarrow} \alpha|001\rangle + \beta|110\rangle \rightarrow (\alpha|0\rangle + \beta|1\rangle)\,|01\rangle \rightarrow (\alpha|0\rangle + \beta|1\rangle)\,|01\rangle$$

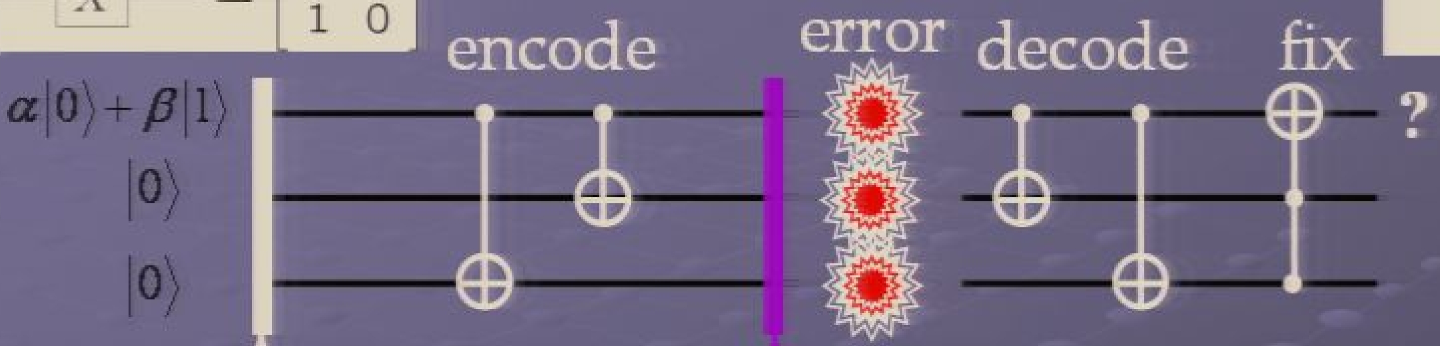
Bit flip Circuit



I = $\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$
X = $\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$



= $\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$



$$(\alpha|0\rangle + \beta|1\rangle) \otimes |0\rangle \otimes |0\rangle \xrightarrow{U} \alpha|0\rangle \otimes |0\rangle \otimes |0\rangle + \beta|1\rangle \otimes |1\rangle \otimes |1\rangle$$

$$\alpha|000\rangle + \beta|111\rangle \xrightarrow{\text{XII}} \alpha|100\rangle + \beta|011\rangle \to (\alpha|1\rangle + \beta|0\rangle)\,|11\rangle \to (\alpha|0\rangle + \beta|1\rangle)\,|11\rangle$$

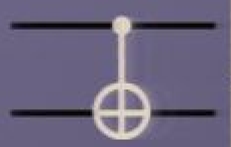
$$\alpha|000\rangle + \beta|111\rangle \xrightarrow{\text{IXI}} \alpha|010\rangle + \beta|101\rangle \to (\alpha|0\rangle + \beta|1\rangle)\,|10\rangle \to (\alpha|0\rangle + \beta|1\rangle)\,|10\rangle$$

$$\alpha|000\rangle + \beta|111\rangle \xrightarrow{\text{IIX}} \alpha|001\rangle + \beta|110\rangle \to (\alpha|0\rangle + \beta|1\rangle)\,|01\rangle \to (\alpha|0\rangle + \beta|1\rangle)\,|01\rangle$$

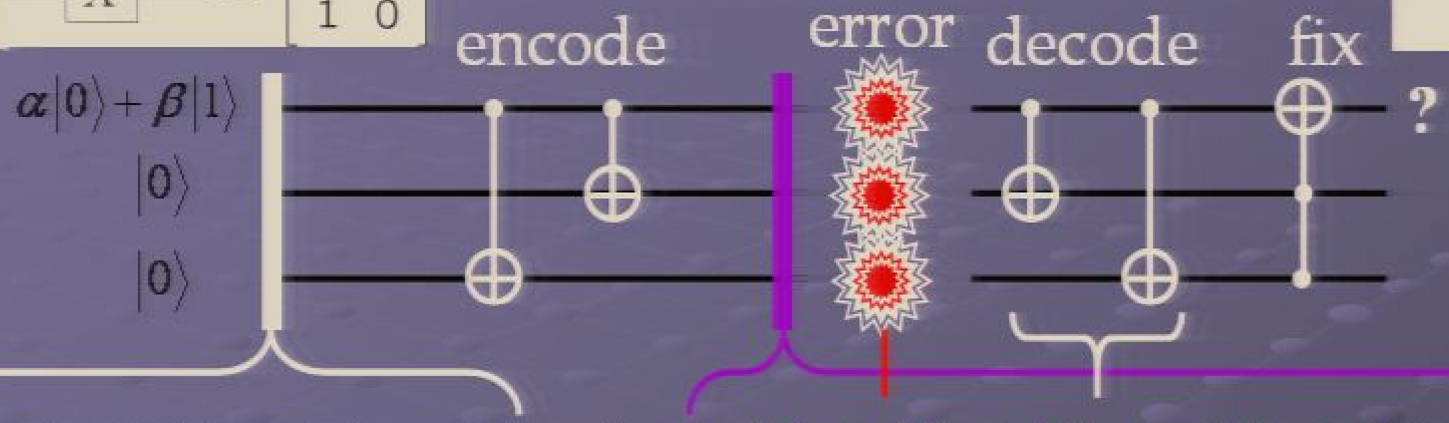
Bit flip Circuit

$\boxed{I} = \begin{bmatrix}1&0\\0&1\end{bmatrix}$

$\boxed{X} = \begin{bmatrix}0&1\\1&0\end{bmatrix}$



$= \begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&0&1\\0&0&1&0\end{bmatrix}$



$$(\alpha|0\rangle+\beta|1\rangle)\otimes|0\rangle\otimes|0\rangle \xrightarrow{U} \alpha|0\rangle\otimes|0\rangle\otimes|0\rangle+\beta|1\rangle\otimes|1\rangle\otimes|1\rangle$$

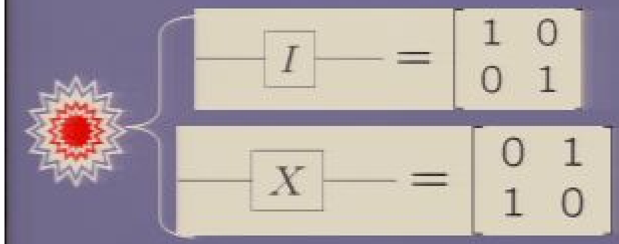
$$\alpha|000\rangle+\beta|111\rangle \overset{XII}{\rightarrow} \alpha|100\rangle+\beta|011\rangle \rightarrow (\alpha|1\rangle+\beta|0\rangle)\,|11\rangle \rightarrow (\alpha|0\rangle+\beta|1\rangle)\,|11\rangle$$

$$\alpha|000\rangle+\beta|111\rangle \overset{IXI}{\rightarrow} \alpha|010\rangle+\beta|101\rangle \rightarrow (\alpha|0\rangle+\beta|1\rangle)\,|10\rangle \rightarrow (\alpha|0\rangle+\beta|1\rangle)\,|10\rangle$$

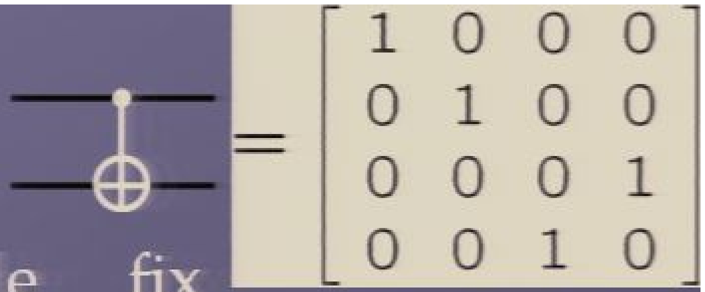
$$\alpha|000\rangle+\beta|111\rangle \overset{IIX}{\rightarrow} \alpha|001\rangle+\beta|110\rangle \rightarrow (\alpha|0\rangle+\beta|1\rangle)\,|01\rangle \rightarrow (\alpha|0\rangle+\beta|1\rangle)\,|01\rangle$$

error decode

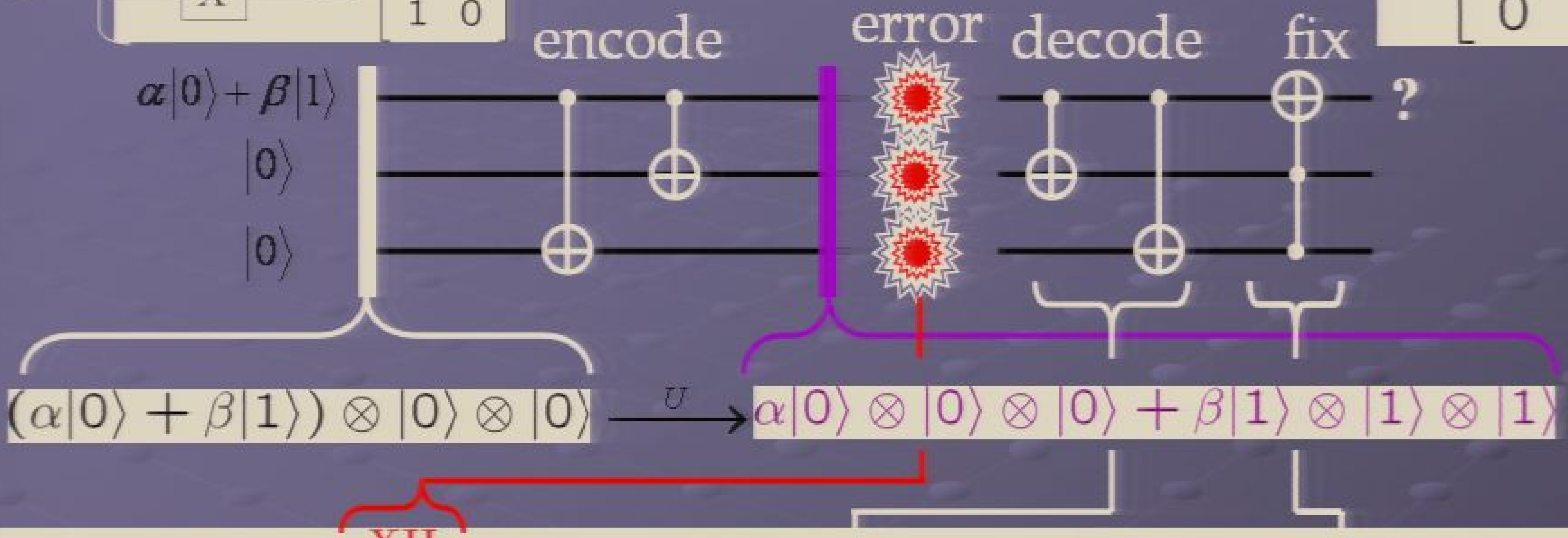
Bit flip Circuit



I = $\begin{bmatrix}1&0\\0&1\end{bmatrix}$
X = $\begin{bmatrix}0&1\\1&0\end{bmatrix}$



= $\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&0&1\\0&0&1&0\end{bmatrix}$



$$(\alpha|0\rangle+\beta|1\rangle)\otimes|0\rangle\otimes|0\rangle \xrightarrow{U} \alpha|0\rangle\otimes|0\rangle\otimes|0\rangle+\beta|1\rangle\otimes|1\rangle\otimes|1\rangle$$

$$\alpha|000\rangle+\beta|111\rangle \xrightarrow{XII} \alpha|100\rangle+\beta|011\rangle \rightarrow (\alpha|1\rangle+\beta|0\rangle)|11\rangle \rightarrow (\alpha|0\rangle+\beta|1\rangle)|11\rangle$$

$$\alpha|000\rangle+\beta|111\rangle \xrightarrow{IXI} \alpha|010\rangle+\beta|101\rangle \rightarrow (\alpha|0\rangle+\beta|1\rangle)|10\rangle \rightarrow (\alpha|0\rangle+\beta|1\rangle)|10\rangle$$

$$\alpha|000\rangle+\beta|111\rangle \xrightarrow{IIX} \alpha|001\rangle+\beta|110\rangle \rightarrow (\alpha|0\rangle+\beta|1\rangle)|01\rangle \rightarrow (\alpha|0\rangle+\beta|1\rangle)|01\rangle$$

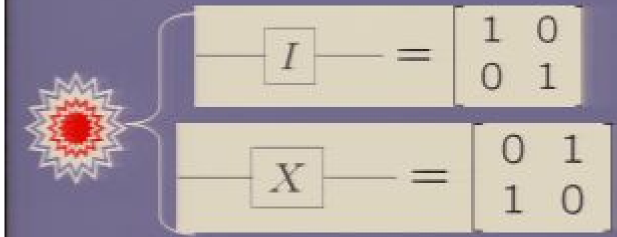
error decode fix

1. encoded into subspace:

$|0\rangle \rightarrow |000\rangle, |1\rangle \rightarrow |111\rangle$

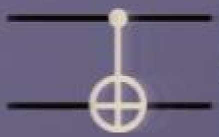
(no-cloning evaded!)

Bit flip Circuit

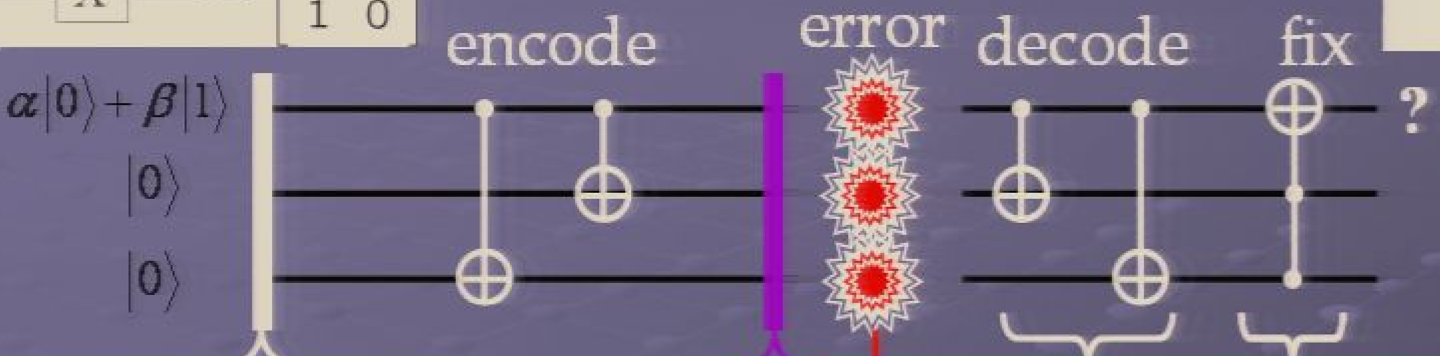


$$= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$



$$= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$



$$(\alpha|0\rangle + \beta|1\rangle) \otimes |0\rangle \otimes |0\rangle \xrightarrow{U} \alpha|0\rangle \otimes |0\rangle \otimes |0\rangle + \beta|1\rangle \otimes |1\rangle \otimes |1\rangle$$

$$\alpha|000\rangle + \beta|111\rangle \overset{XII}{\rightarrow} \alpha|100\rangle + \beta|011\rangle \rightarrow (\alpha|1\rangle + \beta|0\rangle)\,|11\rangle \rightarrow (\alpha|0\rangle + \beta|1\rangle)\,|11\rangle$$

$$\alpha|000\rangle + \beta|111\rangle \overset{IXI}{\rightarrow} \alpha|010\rangle + \beta|101\rangle \rightarrow (\alpha|0\rangle + \beta|1\rangle)\,|10\rangle \rightarrow (\alpha|0\rangle + \beta|1\rangle)\,|10\rangle$$

$$\alpha|000\rangle + \beta|111\rangle \overset{IIX}{\rightarrow} \alpha|001\rangle + \beta|110\rangle \rightarrow (\alpha|0\rangle + \beta|1\rangle)\,|01\rangle \rightarrow (\alpha|0\rangle + \beta|1\rangle)\,|01\rangle$$

error decode fix

1. encoded into subspace:

   $|0\rangle \rightarrow |000\rangle, |1\rangle \rightarrow |111\rangle$

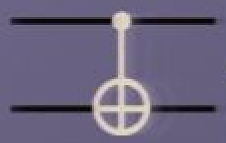
   (no-cloning evaded!)

2. errors take to orthogonal subspaces + maintain orthogonality

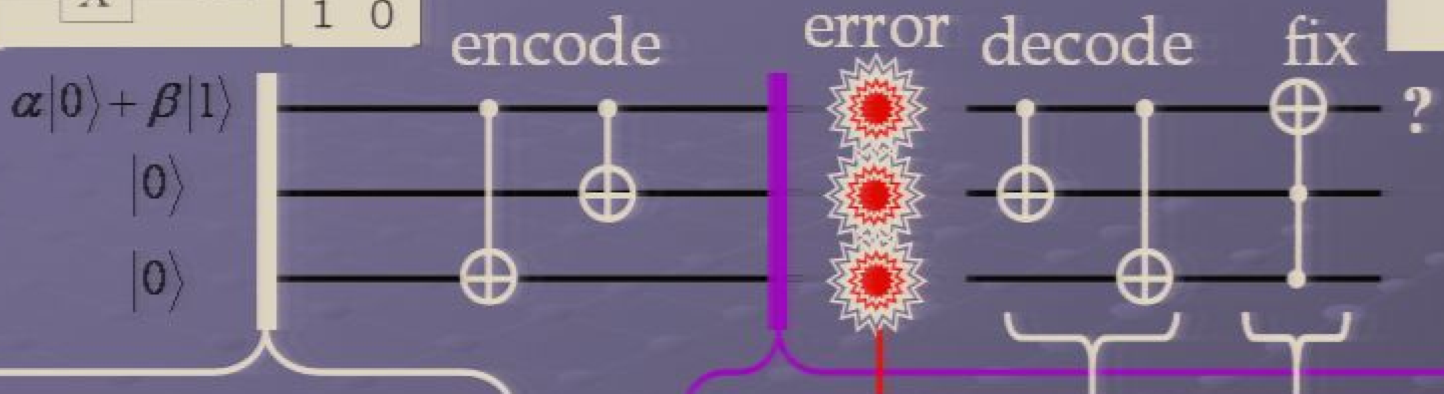
   $XII|000\rangle = |100\rangle, XII|111\rangle = |011\rangle$

Bit flip Circuit

$$-\boxed{I}- = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \qquad -\boxed{X}- = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$



$$= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$



$$(\alpha|0\rangle+\beta|1\rangle)\otimes|0\rangle\otimes|0\rangle \xrightarrow{U} \alpha|0\rangle\otimes|0\rangle\otimes|0\rangle+\beta|1\rangle\otimes|1\rangle\otimes|1\rangle$$

$$\alpha|000\rangle+\beta|111\rangle \overset{XII}{\rightarrow} \alpha|100\rangle+\beta|011\rangle \rightarrow (\alpha|1\rangle+\beta|0\rangle)\,|11\rangle \rightarrow (\alpha|0\rangle+\beta|1\rangle)\,|11\rangle$$

$$\alpha|000\rangle+\beta|111\rangle \overset{IXI}{\rightarrow} \alpha|010\rangle+\beta|101\rangle \rightarrow (\alpha|0\rangle+\beta|1\rangle)\,|10\rangle \rightarrow (\alpha|0\rangle+\beta|1\rangle)\,|10\rangle$$

$$\alpha|000\rangle+\beta|111\rangle \overset{IIX}{\rightarrow} \alpha|001\rangle+\beta|110\rangle \rightarrow (\alpha|0\rangle+\beta|1\rangle)\,|01\rangle \rightarrow (\alpha|0\rangle+\beta|1\rangle)\,|01\rangle$$

error decode fix

1. encoded into subspace: $|0\rangle \rightarrow |000\rangle, |1\rangle \rightarrow |111\rangle$ (no-cloning evaded!)

2. errors take to orthogonal subspaces + maintain orthogonality $XII|000\rangle = |100\rangle, XII|111\rangle = |011\rangle$

3. syndrome

Bit flip *a*

Bit flip and Phase flip errors

$$\left\{\begin{array}{l} -\boxed{I}- = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \\ -\boxed{Z}- = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \end{array}\right.$$

Bit flip and Phase flip errors

$$-\boxed{I}- = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$-\boxed{Z}- = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

$$Z|0\rangle = |0\rangle$$

$$Z|1\rangle = -|1\rangle$$

Bit flip and Phase flip errors

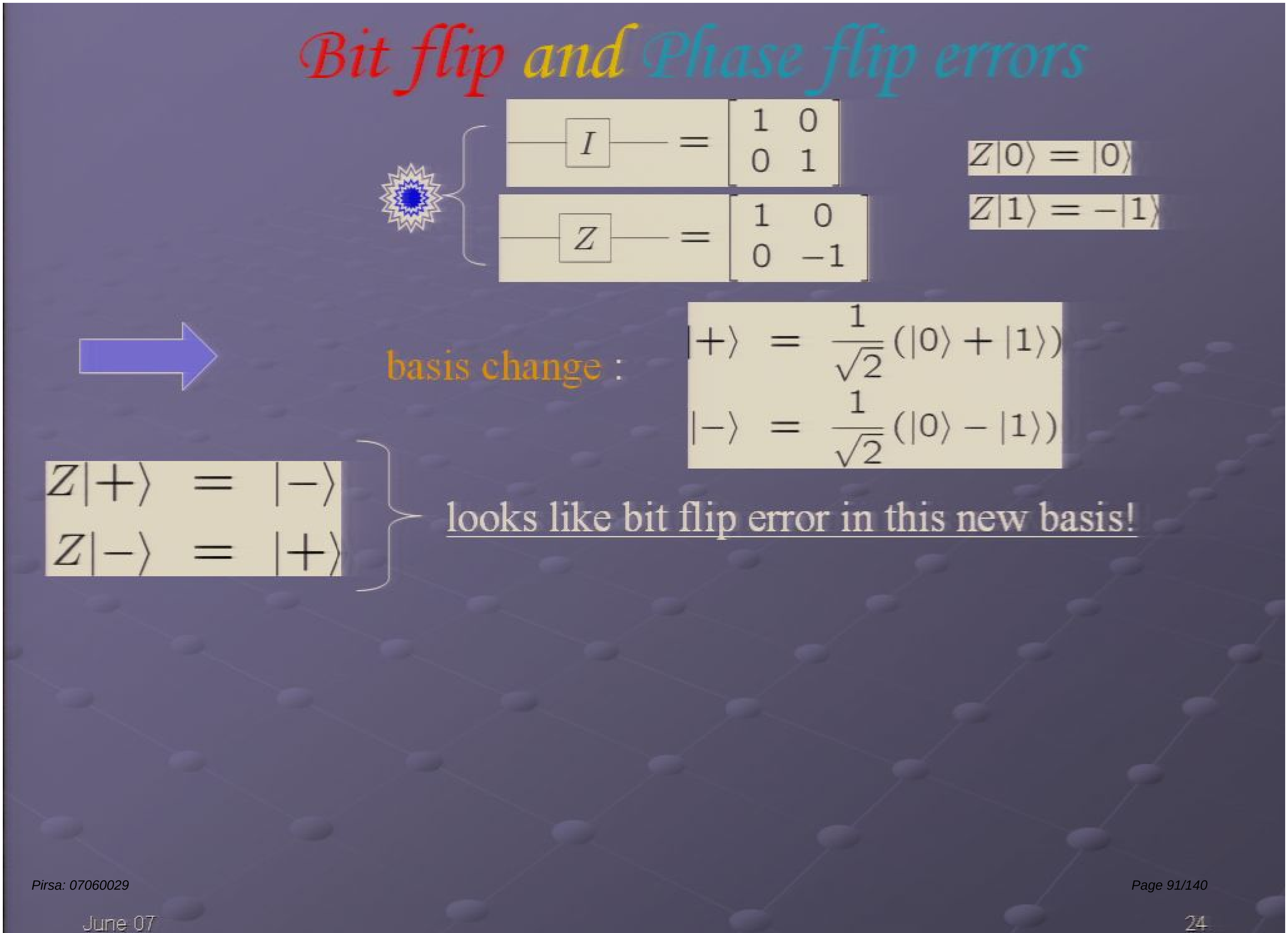
$$\left\{\begin{array}{l} -\boxed{I}- \;=\; \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \\[1em] -\boxed{Z}- \;=\; \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \end{array}\right.$$

$$Z|0\rangle = |0\rangle$$

$$Z|1\rangle = -|1\rangle$$

basis change :

$$\begin{aligned} |+\rangle &= \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \\ |-\rangle &= \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \end{aligned}$$



Bit flip and Phase flip errors

$$-\boxed{I}- = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$-\boxed{Z}- = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

$$Z|0\rangle = |0\rangle$$

$$Z|1\rangle = -|1\rangle$$

basis change :

$$\begin{aligned} |+\rangle &= \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \\ |-\rangle &= \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \end{aligned}$$

$$\left.\begin{aligned} Z|+\rangle &= |-\rangle \\ Z|-\rangle &= |+\rangle \end{aligned}\right\}$$

looks like bit flip error in this new basis!

Bit flip and Phase flip errors

$$-\boxed{I}- = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$-\boxed{Z}- = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

$Z|0\rangle = |0\rangle$

$Z|1\rangle = -|1\rangle$

basis change :

$$\begin{aligned} |+\rangle &= \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \\ |-\rangle &= \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \end{aligned}$$

$$\left.\begin{aligned} Z|+\rangle &= |-\rangle \\ Z|-\rangle &= |+\rangle \end{aligned}\right\}$$

looks like bit flip error in this new basis!

$$\boxed{HZH = X, \quad H = |+\rangle\langle 0| + |-\rangle\langle 1}$$

Bit flip and Phase flip errors

$$-\boxed{I}- = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$-\boxed{Z}- = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

$Z|0\rangle = |0\rangle$

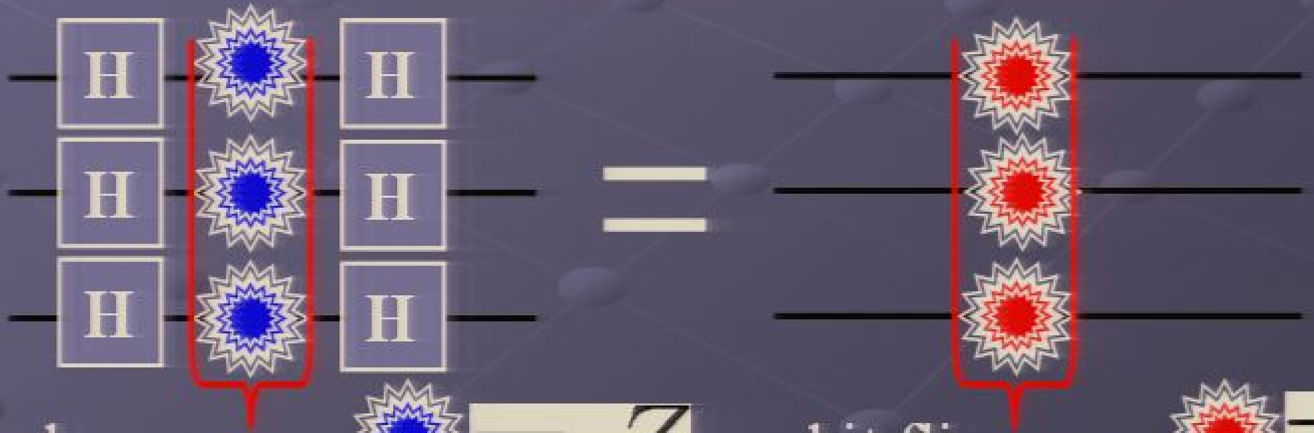
$Z|1\rangle = -|1\rangle$

basis change :
$$\begin{aligned} |+\rangle &= \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \\ |-\rangle &= \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \end{aligned}$$

$$\begin{aligned} Z|+\rangle &= |-\rangle \\ Z|-\rangle &= |+\rangle \end{aligned}$$

looks like bit flip error in this new basis!

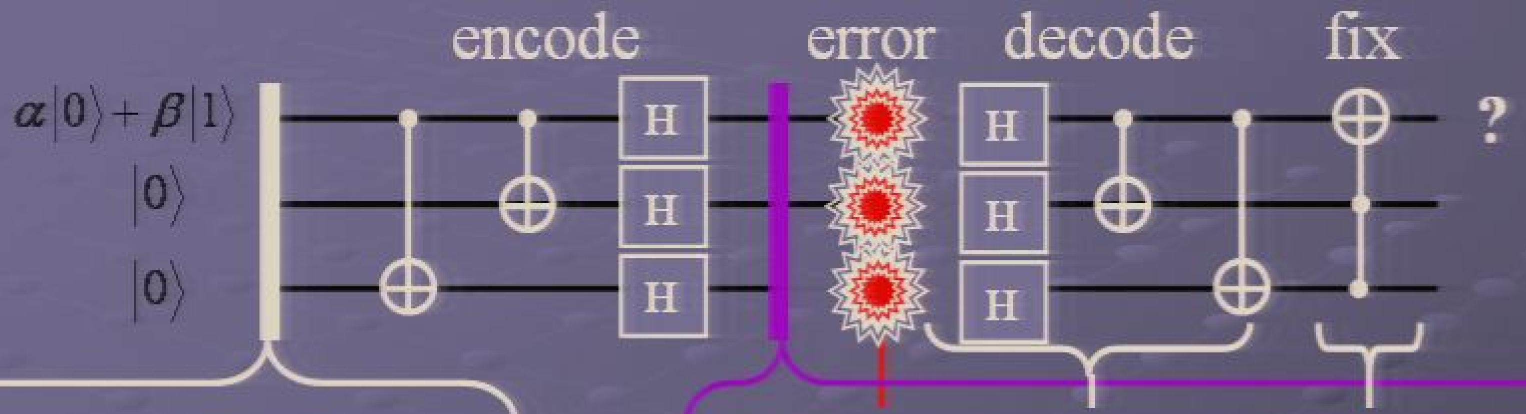
$$\boxed{HZH = X, \quad H = |+\rangle\langle 0| + |-\rangle\langle 1}$$



phase errors $= Z$ bit flip errors $= X$

Phase errors cir

Phase errors circuit



$$(\alpha|0\rangle+\beta|1\rangle)\otimes|0\rangle\otimes|0\rangle \xrightarrow{U} \alpha|+\rangle\otimes|+\rangle\otimes|+\rangle+\beta|-\rangle\otimes|-\rangle\otimes|-\rangle$$

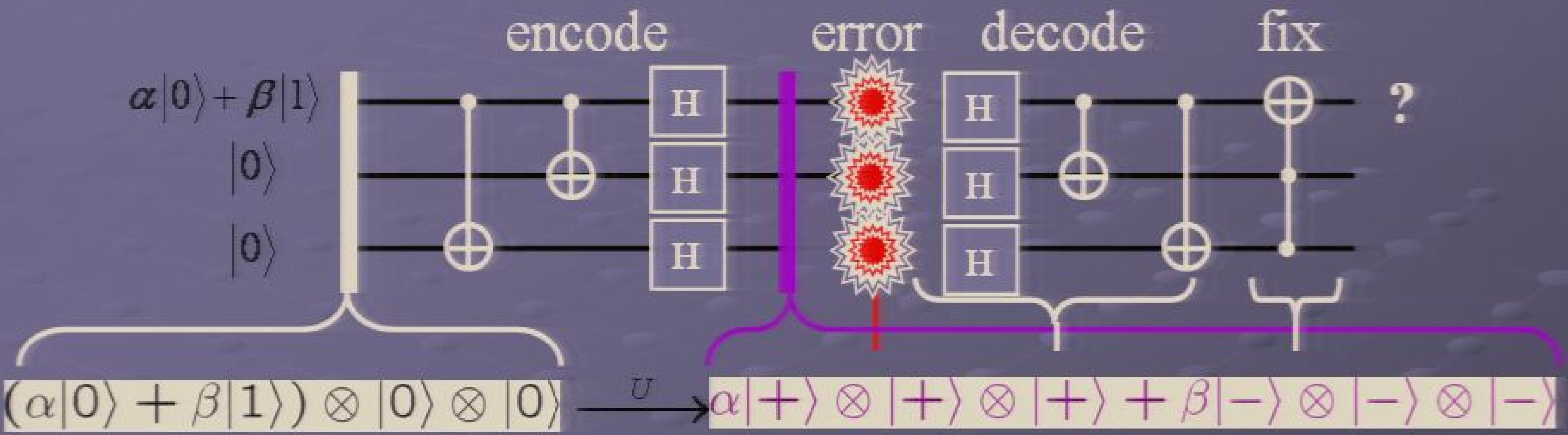
$$\alpha|+++\rangle+\beta|---\rangle \overset{ZII}{\rightarrow} \alpha|-++\rangle+\beta|+--\rangle \rightarrow (\alpha|1\rangle+\beta|0\rangle)|11\rangle \rightarrow (\alpha|0\rangle+\beta|1\rangle)|11\rangle$$

$$\alpha|+++\rangle+\beta|---\rangle \overset{IZI}{\rightarrow} \alpha|+-+\rangle+\beta|-+-\rangle \rightarrow (\alpha|0\rangle+\beta|1\rangle)|10\rangle \rightarrow (\alpha|0\rangle+\beta|1\rangle)|10\rangle$$

$$\alpha|+++\rangle+\beta|---\rangle \overset{IIZ}{\rightarrow} \alpha|++-\rangle+\beta|--+\rangle \rightarrow (\alpha|0\rangle+\beta|1\rangle)|01\rangle \rightarrow (\alpha|0\rangle+\beta|1\rangle)|01\rangle$$

error decode fix

Phase errors circuit



$$(\alpha|0\rangle+\beta|1\rangle)\otimes|0\rangle\otimes|0\rangle \xrightarrow{U} \alpha|+\rangle\otimes|+\rangle\otimes|+\rangle+\beta|-\rangle\otimes|-\rangle\otimes|-\rangle$$

$$\alpha|+++\rangle+\beta|---\rangle \overset{ZII}{\rightarrow} \alpha|-++\rangle+\beta|+--\rangle \rightarrow (\alpha|1\rangle+\beta|0\rangle)|11\rangle \rightarrow (\alpha|0\rangle+\beta|1\rangle)|11\rangle$$

$$\alpha|+++\rangle+\beta|---\rangle \overset{IZI}{\rightarrow} \alpha|+-+\rangle+\beta|-+-\rangle \rightarrow (\alpha|0\rangle+\beta|1\rangle)|10\rangle \rightarrow (\alpha|0\rangle+\beta|1\rangle)|10\rangle$$

$$\alpha|+++\rangle+\beta|---\rangle \overset{IIZ}{\rightarrow} \alpha|++-\rangle+\beta|--+\rangle \rightarrow (\alpha|0\rangle+\beta|1\rangle)|01\rangle \rightarrow (\alpha|0\rangle+\beta|1\rangle)|01\rangle$$

error decode fix

1. encoded into subspace:

$|0\rangle \rightarrow |+++\rangle, |1\rangle \rightarrow |---\rangle$

(no-cloning evaded!)

2. errors take to orthogonal subspaces + maintain orthogonality

$ZII|+++\rangle = |-++\rangle, ZII|---\rangle = |+--\rangle$

3. syndrome

Shor's

Shor's code summary

3 qubit bit flip code

$$|0\rangle \to |000\rangle$$
$$|1\rangle \to |111\rangle$$

3 qubit phase flip code

$$|0\rangle \to |+++\rangle$$
$$|1\rangle \to |---\rangle$$

Shor's code summary

3 qubit bit flip code

$$\begin{aligned}|0\rangle &\rightarrow |000\rangle\\ |1\rangle &\rightarrow |111\rangle\end{aligned}$$

3 qubit phase flip code

$$\begin{aligned}|0\rangle &\rightarrow |+++\rangle\\ |1\rangle &\rightarrow |---\rangle\end{aligned}$$

phase errors $\mathbf{ZII},\ \mathbf{IZI},\ \mathbf{IIZ}$ act as $\mathbf{Z}$ on bit flip code qubits:

$$\begin{aligned}|0\rangle_B &= |000\rangle\\ |1\rangle_B &= |111\rangle\end{aligned} \qquad \begin{aligned}\mathbf{ZII}|0\rangle_B &= |0\rangle_B\\ \mathbf{ZII}|1\rangle_B &= -|1\rangle_B\end{aligned}$$

Shor's code summary

3 qubit bit flip code

$$\begin{aligned}|0\rangle &\rightarrow |000\rangle\\ |1\rangle &\rightarrow |111\rangle\end{aligned}$$

3 qubit phase flip code

$$\begin{aligned}|0\rangle &\rightarrow |+++\rangle\\ |1\rangle &\rightarrow |---\rangle\end{aligned}$$

phase errors **ZII, IZI, IIZ** act as **Z** on bit flip code qubits:

$$\begin{aligned}|0\rangle_B &= |000\rangle & \mathbf{ZII}|0\rangle_B &= |0\rangle_B\\ |1\rangle_B &= |111\rangle & \mathbf{ZII}|1\rangle_B &= -|1\rangle_B\end{aligned}$$

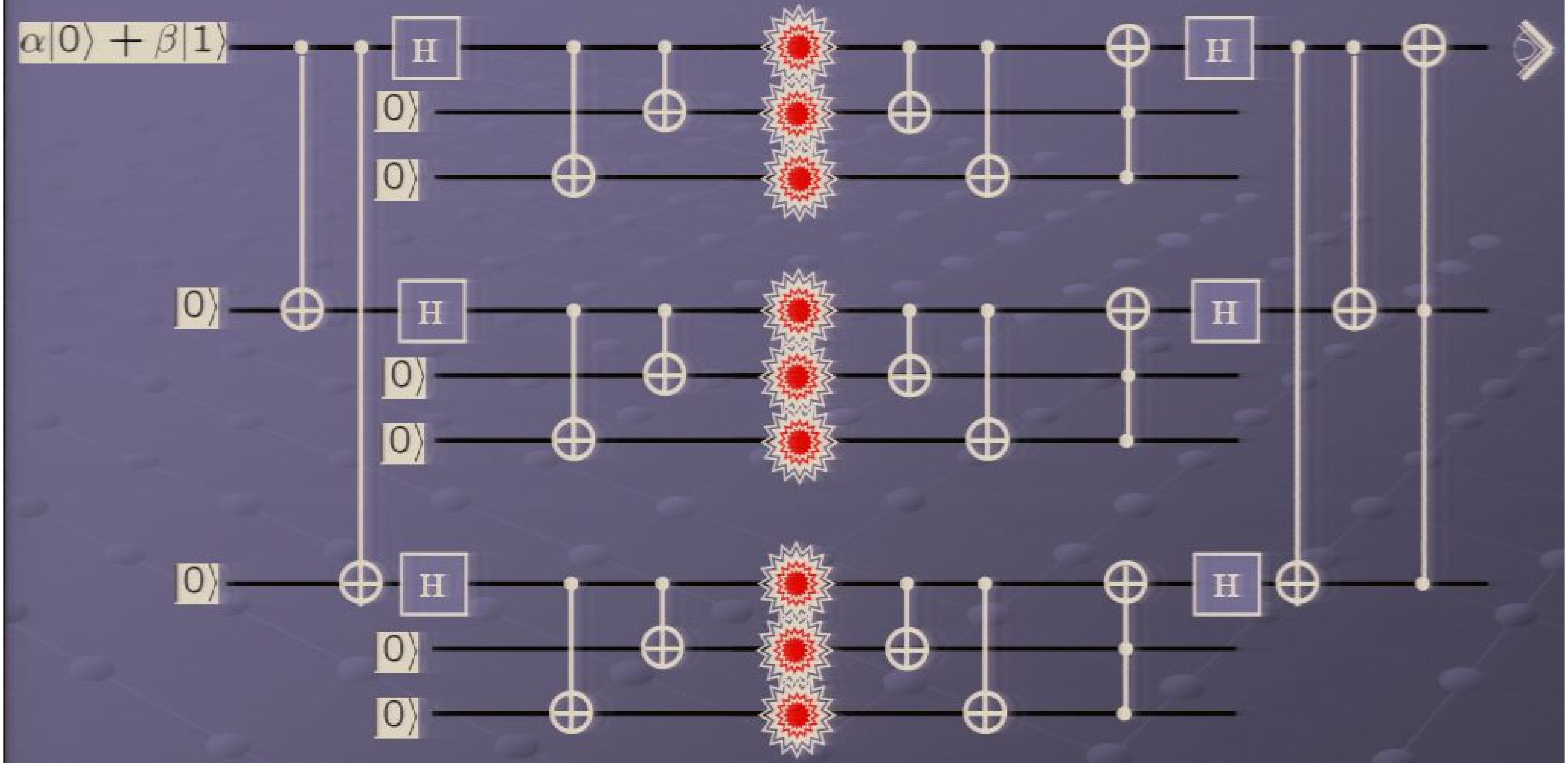
define:

$$\begin{aligned}|p\rangle &= \frac{1}{\sqrt{2}}(|0\rangle_B + |1\rangle_B) = \frac{1}{\sqrt{2}}(|000\rangle + |111\rangle)\\ |m\rangle &= \frac{1}{\sqrt{2}}(|0\rangle_B - |1\rangle_B) = \frac{1}{\sqrt{2}}(|000\rangle - |111\rangle)\end{aligned}$$

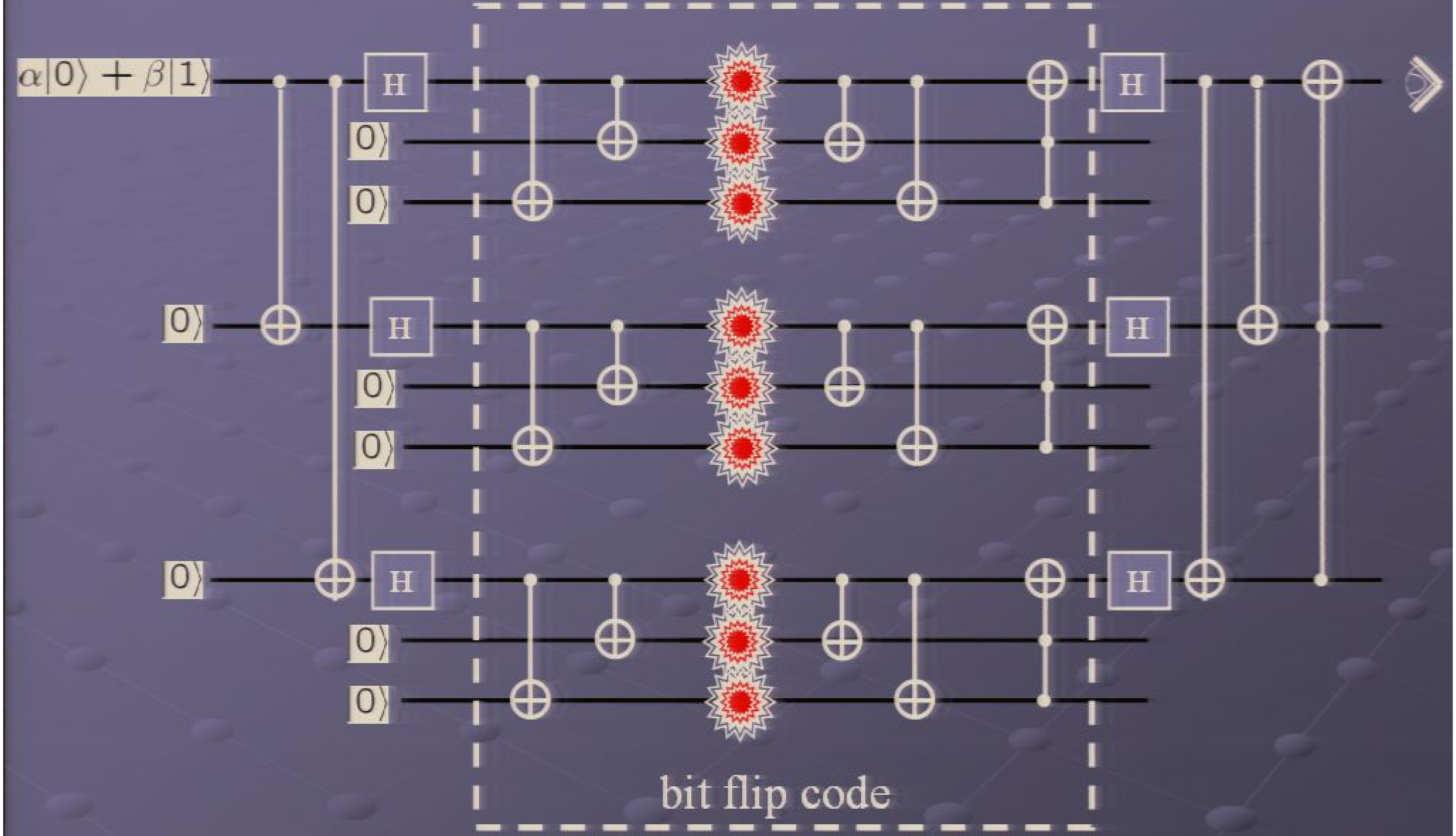
Shor Code: (Peter Shor, 1995)

$$\begin{aligned}|0\rangle &\rightarrow |ppp\rangle = \frac{1}{2\sqrt{2}}(|000\rangle + |111\rangle)(|000\rangle + |111\rangle)(|000\rangle + |111\rangle)\\ |1\rangle &\rightarrow |mmm\rangle = \frac{1}{2\sqrt{2}}(|000\rangle - |111\rangle)(|000\rangle - |111\rangle)(|000\rangle - |111\rangle)\end{aligned}$$

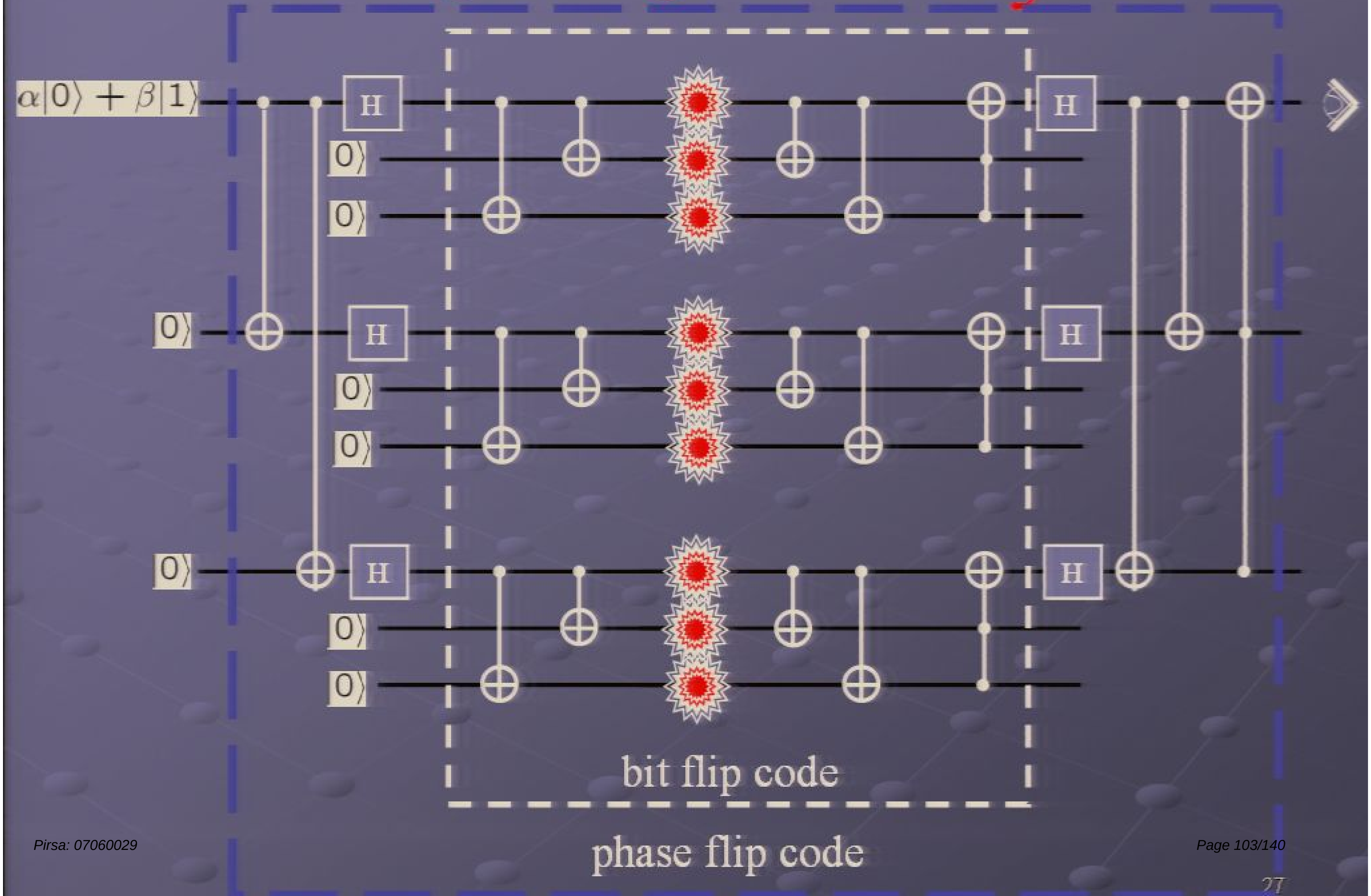
Shor's Code circuitry



Shor's Code circuitry



Shor's Code circuitry



Fault Tolerant Quantum Computation

Fault Tolerant Quantum computa

Fault Tolerant Quantum computation

- Fault Tolerance computation requires that if the probability of introducing error in the circuit is p, the probability that the circuit brings two or more errors grows like $O(p^2)$ This means that a fault tolerant procedure comes to end succesfully with $1-cp^2$ probability where c depends only on the circuit

Fault Tolerant Quantum computation

- Fault Tolerance computation requires that if the probability of introducing error in the circuit is p, the probability that the circuit brings two or more errors grows like $O(p^2)$ This means that a fault tolerant procedure comes to end succesfully with 1-cp^2 probability where c depends only on the circuit
- Concatenation methods brings a square benefit in reducing error probability decreasing the factor from cp^2 to $c(cp^2)^2$

Fault Tolerant Quantum computation

- Fault Tolerance computation requires that if the probability of introducing error in the circuit is p, the probability that the circuit brings two or more errors grows like $O(p^2)$ This means that a fault tolerant procedure comes to end succesfully with $1\text{-}cp^2$ probability where c depends only on the circuit
- Concatenation methods brings a square benefit in reducing error probability decreasing the factor from cp^2 to $c(cp^2)^2$
- Moreover if the probability p maintain its value under this

$$p < cp^2 \Rightarrow p_{fail\,(threshold)} < \frac{1}{c}$$

Fault Tolerant Quantum computation

- Fault Tolerance computation requires that if the probability of introducing error in the circuit is p, the probability that the circuit brings two or more errors grows like $O(p^2)$ This means that a fault tolerant procedure comes to end succesfully with 1-cp² probability where c depends only on the circuit
- Concatenation methods brings a square benefit in reducing error probability decreasing the factor from cp^2 to $c(cp^2)^2$
- Moreover if the probability p maintain its value under this

$$p < cp^2 \Rightarrow p_{fail\,(threshold)} < \frac{1}{c}$$

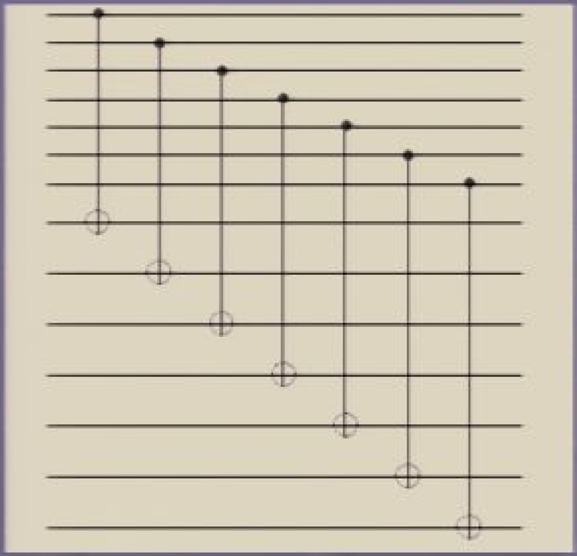
- Arbitrary accuracy could be reached with a dimensional growth that scales as

$$d^a = O\left(poly\left(\log\left(\frac{1}{c\varepsilon}\right)\right)\right)$$

Threshold

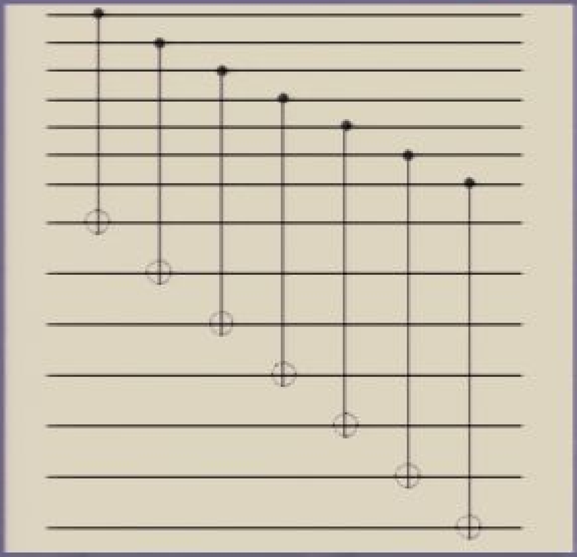
Example: Fault toler

Example: *Fault tolerant CNOT*



7 qubit transversal Cnot

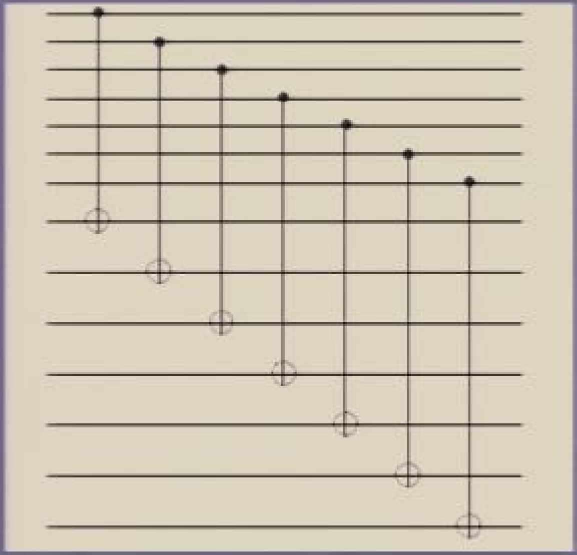
Example: Fault tolerant CNOT



7 qubit transversal Cnot

Fault tolerant rules requires not to introduce more than one error for every encoded block
(Transversal gates)

Example: Fault tolerant CNOT

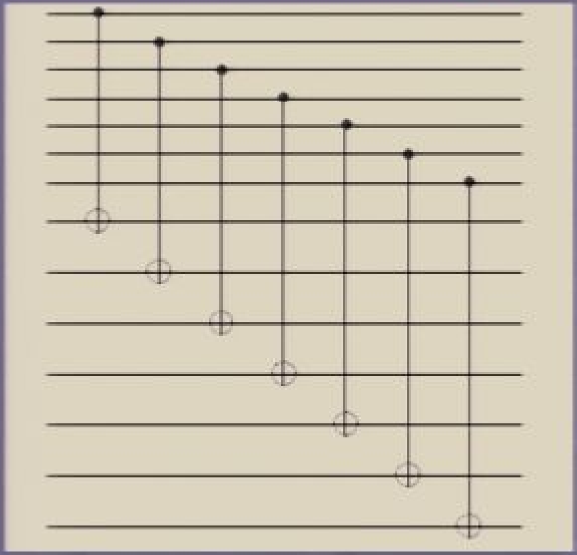


7 qubit transversal Cnot

Fault tolerant rules requires not to introduce more than one error for every encoded block (Transversal gates)

Beside Cnot it's

Example: Fault tolerant CNOT

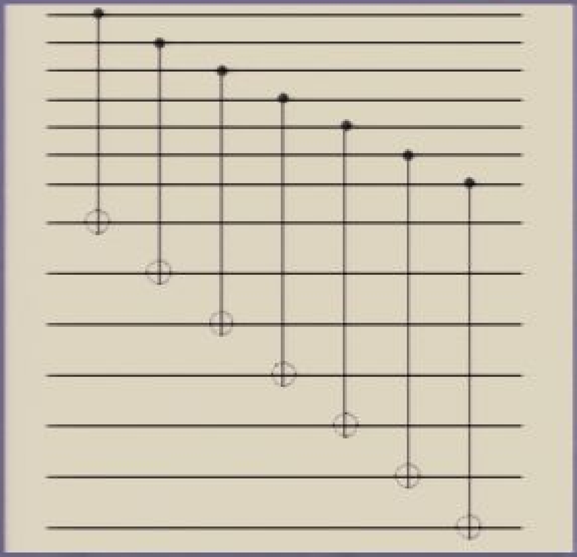


7 qubit transversal Cnot

Fault tolerant rules requires not to introduce more than one error for every encoded block (Transversal gates)

Beside Cnot it's possible to realize on the encoded states a un

Example: *Fault tolerant CNOT*

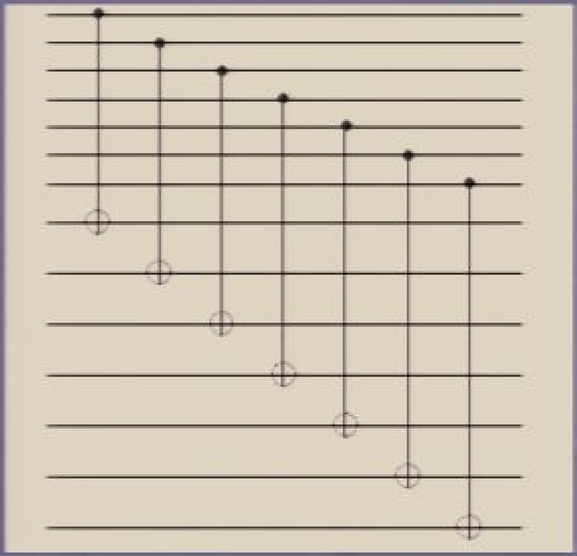


7 qubit transversal Cnot

Fault tolerant rules requires not to introduce more than one error for every encoded block
(Transversal gates)

Beside Cnot it's possible to realize on the encoded states a universal gate set

Example: Fault tolerant CNOT



7 qubit transversal Cnot

Fault tolerant rules requires not to introduce more than one error for every encoded block (Transversal gates)

Beside Cnot it's possible to realize on the encoded states a universal gate set

Gottesman-Knill theorem has demonstrated the efficient simulation on a classical computer of quantum computation with fault tolerant quantum gates (Clifford group gates: Hadamard, Phase, CNOT).

Subsystem Coding

Subsystem Coding

Subsystem Coding

- Recently it was realized that, since the most general method for encoding quantum information is to encode it into a subsystem, there exists a novel form of quantum error correction beyond the traditional quantum error correcting subspace codes
- These new quantum error correcting subsystem codes differ from subspace codes in that their quantum correcting routines can be considerably simpler than related subspace codes

Subsystem Coding

- Recently it was realized that, since the most general method for encoding quantum information is to encode it into a subsystem, there exists a novel form of quantum error correction beyond the traditional quantum error correcting subspace codes
- These new quantum error correcting subsystem codes differ from subspace codes in that their quantum correcting routines can be considerably simpler than related subspace codes

Subsystem Space

$$H = (D \otimes C) \oplus \mathcal{E}$$

Code Space

$$H = \bigoplus_{s_1^X, \dots, s_{n-1}^X, s_1^Z, \dots, s_{n-1}^Z = \pm 1} H^T_{s^X, s^Z} \otimes H^L_{s^X, s^Z}$$

Subsystem Error Correction

We only need to protect subsystem, not the full subspaces.

Subsystem Error Correction

We only need to protect subsystem, not the full subspaces.

$\mathcal{H}_{sys} = (\mathcal{C} \otimes \mathcal{D}) \oplus \mathcal{P}$

encoding subsystem

$$(|\phi\rangle\langle\phi| \otimes \rho_D) \oplus 0 \;-\; \boxed{\mathcal{E}} \;-\; \boxed{\mathcal{R}} \;-\; (|\phi\rangle\langle\phi| \otimes \tilde{\rho}_D) \oplus 0$$

Error Recovery

Subsystem Error Correction

We only need to protect subsystem, not the full subspaces.

$$\mathcal{H}_{sys} = (\mathcal{C} \otimes \mathcal{D}) \oplus \mathcal{P}$$

encoding subsystem

$$\begin{tikzcd} (|\phi\rangle\langle\phi| \otimes \rho_D) \oplus 0 \arrow[r, dash] & \boxed{\mathcal{E}} \arrow[r, dash] & \boxed{\mathcal{R}} \arrow[r, dash] & (|\phi\rangle\langle\phi| \otimes \tilde{\rho}_D) \oplus 0 \end{tikzcd}$$

Error Recovery

encoded quantum information

Correction conditions

$$\langle \phi_i | E_i^\dagger E_j | \phi_j \rangle = C_{ij} \delta_{ij} \Rightarrow \langle \phi_i | \otimes \langle \phi_k | E_i^\dagger E_j | \phi_j \rangle \otimes | \phi_l \rangle = C_{ijkl} \delta_{ij}$$

V_z^n :

$|0\rangle_L = |00..0\rangle$

$|1\rangle_L = |11..1\rangle$

$S_i : ZZI, ZIZ, IZZ$

An example:
$[n^2, 1, n]$ Shor Code

- For a n redundancy code the stabilizers are all pairs of Z acting on every pair of qubits among the n qubits in the code block. Similarly for the redundance code in the Hadamard rotated basis, the stabilizers are all possible pairs of X acting on n qubits.

V_z^n :

$|0\rangle_L = |00..0\rangle$

$|1\rangle_L = |11..1\rangle$

$S_i : ZZI, ZIZ, IZZ$

An example:

$[n^2, 1, n]$ Shor Code

V_x^n :

$|+\rangle_L = |++..+\rangle$

$|-\rangle_L = |--..-\rangle$

$S_i : XXI, XIX, IXX$

- For a n redundancy code the stabilizers are all pairs of Z acting on every pair of qubits among the n qubits in the code block. Similarly for the redundance code in the Hadamard rotated basis, the stabilizers are all possible pairs of X acting on n qubits.
- For the Shor concatenated code the stabilizers on each code sub block are the same for the redundancy code: i.e. pairs of Z operators acting on every pairs of qubit in any given sub block.
- In addition because of the Hadamard rotated basis encoding we have stabilizers which are pairs of encoded X operators acting on every pair of sub-blocks in the code block
- An encoded X operator in a sub-block should take 0_L to 1_L and viceversa and this could be accomplished by an X operator acting on all qubits of the sub-block.

V_z^n :
$|0\rangle_L = |00..0\rangle$
$|1\rangle_L = |11..1\rangle$
$S_i : ZZI, ZIZ, IZZ$

An example:

$[n^2, 1, n]$ Shor Code

V_x^n :
$|+\rangle_L = |++..+\rangle$
$|-\rangle_L = |--..-\rangle$
$S_i : XXI, XIX, IXX$

- For a n redundancy code the stabilizers are all pairs of Z acting on every pair of qubits among the n qubits in the code block. Similarly for the redundance code in the Hadamard rotated basis, the stabilizers are all possible pairs of X acting on n qubits.
- For the Shor concatenated code the stabilizers on each code sub block are the same for the redundancy code: i.e. pairs of Z operators acting on every pairs of qubit in any given sub block.
- In addition because of the Hadamard rotated basis encoding we have stabilizers which are pairs of encoded X operators acting on every pair of sub-blocks in the code block
- An encoded X operator in a sub-block should take 0_L to 1_L and viceversa and this could be accomplished by an X operator acting on all qubits of the sub-block.
- The concatenated code include X operators acting on all qubits on every pair of sub-blocks in the code blocks-

$[n^2,1,n]$ Shor Code (2)

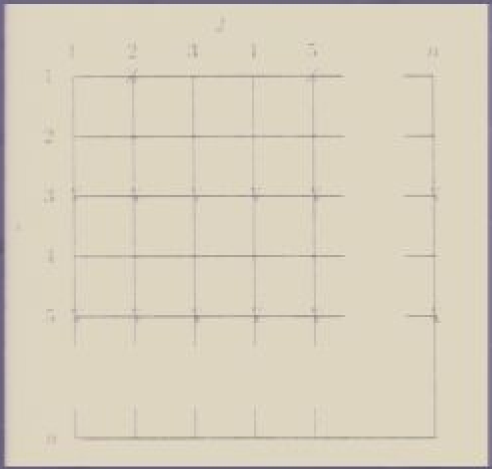
Placing Qubits in different sub-blocks to lie on different rows, the stabilizer includes X operators acting on all qubits in every pair of rows. Furthermore within each sub-block (each row) the stabilizer includes Z operators acting on all pairs of qubits in the corresponding row. Same comments on X basis

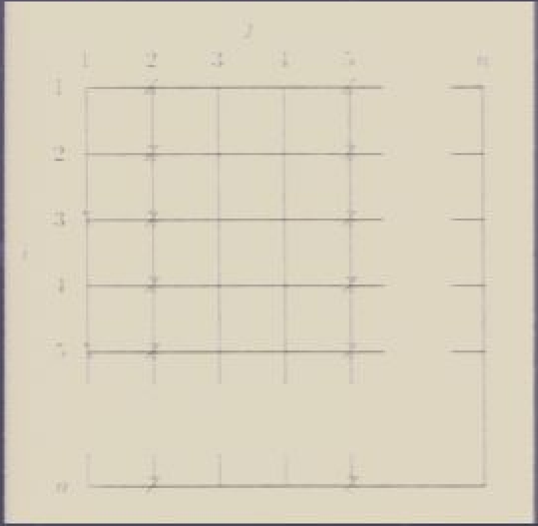
More formally Shor's code is generated by

(n-1)+n(n-1)=n²-1 operators

$S_{C_Z \circ C_x} = \left\langle Z_{col-j,col-(j+1)}; X_{i,j} X_{i+1,j}; X_{i,n} X_{i+1,n} : i,j \in ¢_{n-1} \right\rangle$

$S_{C_Z \circ C_x} = \left\langle Z_{col-j,col-(j+1)}; X_{i,j} X_{i+1,j}; X_{i,n} X_{i+1,n} : i,j \in ¢_{n-1} \right\rangle$





$|k\rangle \propto |+\rangle + (-1)^k |-\rangle$

$V_X^n \to |++\ldots+\rangle + (-1)^k |--\ldots-\rangle$

$V_Z^n \circ V_X^n \to |\bar{+}\bar{+}\ldots\bar{+}\rangle + (-1)^k |\bar{-}\bar{-}\ldots\bar{-}\rangle$

$|\bar{+}\rangle \propto |0\rangle + |1\rangle, \ |\bar{-}\rangle \propto |0\rangle - |1\rangle$

$|k\rangle \propto |0\rangle + (-1)^k |1\rangle$

$V_Z^n \to |00\ldots0\rangle + (-1)^k |11\ldots1\rangle$

$V_X^n \circ V_Z^n \to |\bar{0}\bar{0}\ldots\bar{0}\rangle + (-1)^k |\bar{1}\bar{1}\ldots\bar{1}\rangle$

$|\bar{0}\rangle \propto |+\rangle + |-\rangle, \ |\bar{1}\rangle \propto |+\rangle - |-\rangle$

Z basis

X basis

Bacon-Shor Codes

- **This code is a generalization of Shor's original quantum error correcting subspace code.**
- It eliminates the asymmetry inside Shor Code in the treatment of Z errors and X errors.
- This code is able to correct n/2 X and Z errors and it's generated

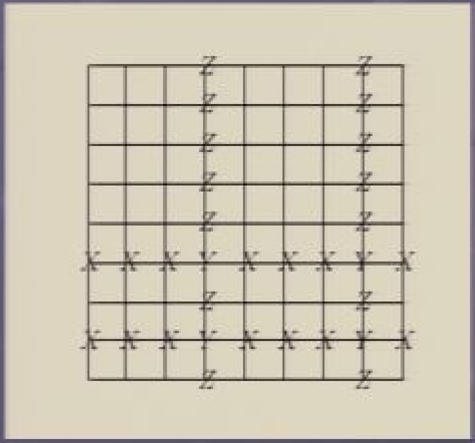
Bacon-Shor Codes

- **This code is a generalization of Shor's original quantum error correcting subspace code.**
- It eliminates the asymmetry inside Shor Code in the treatment of Z errors and X errors.
- This code is able to correct n/2 X and Z errors and it's generated by

Bacon-Shor Codes

- **This code is a generalization of Shor's original quantum error correcting subspace code.**
- It eliminates the asymmetry inside Shor Code in the treatment of Z errors and X errors.
- This code is able to correct n/2 X and Z errors and it's generated by

$$S_{BS} = \left\langle X_{row-i,row-(i+1)}; Z_{col-j,col-(j+1)} : i, j \in ¢_{n-1} \right\rangle$$



$[n^2,1,n]$ Shor Code (2)

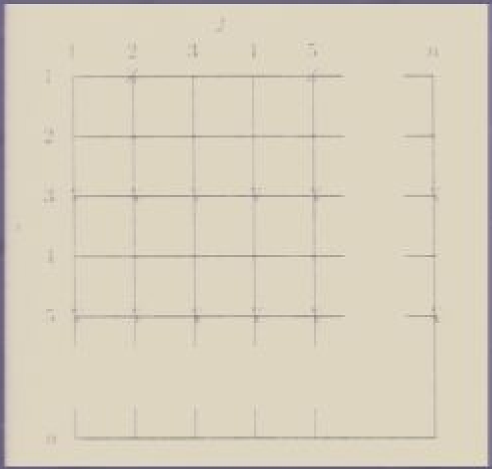
Placing Qubits in different sub-blocks to lie on different rows, the stabilizer includes X operators acting on all qubits in every pair of rows. Furthermore within each sub-block (each row) the stabilizer includes Z operators acting on all pairs of qubits in the corresponding row. Same comments on X basis

More formally Shor's code is generated by

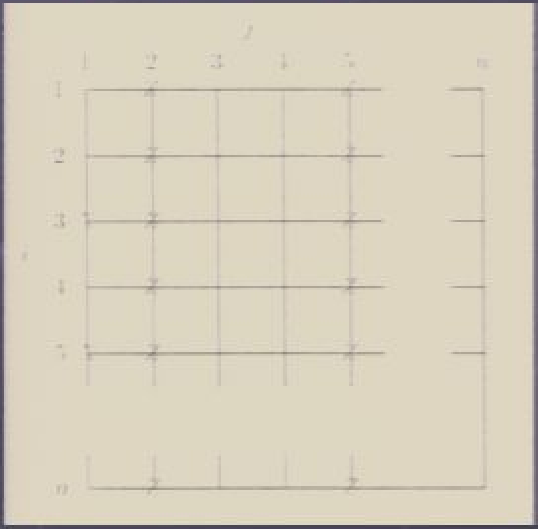
$(n-1)+n(n-1)=n^2-1$ operators

$$S_{C_zoC_x} = \left\langle Z_{col-j,col-(j+1)}; X_{i,j}X_{i+1,j}; X_{i,n}X_{i+1,n} : i,j \in ¢_{n-1} \right\rangle$$

$$S_{C_xoC_z} = \left\langle Z_{col-j,col-(j+1)}; X_{i,j}X_{i+1,j}; X_{i,n}X_{i+1,n} : i,j \in ¢_{n-1} \right\rangle$$



Shor's code construction treats X and Z errors asymmetrically. Within each of the subblocks up to n/2 errors can be corrected because of the underlying Repetition code. The code can also Correct up to n/2 Z errors in different Subblocks (viceversa in the X basis)



$|k\rangle \propto |+\rangle + (-1)^k|-\rangle$

$V_X^n \rightarrow |++\ldots+\rangle + (-1)^k|--\ldots-\rangle$

$V_Z^n o V_X^n \rightarrow |\bar{+}\bar{+}\ldots\bar{+}\rangle + (-1)^k|\bar{-}\bar{-}\ldots\bar{-}\rangle$

$|\bar{+}\rangle \propto |0\rangle + |1\rangle \quad |\bar{-}\rangle \propto |0\rangle - |1\rangle$

$|k\rangle \propto |0\rangle + (-1)^k|1\rangle$

$V_Z^n \rightarrow |00\ldots0\rangle + (-1)^k|11\ldots1\rangle$

$V_X^n o V_Z^n \rightarrow |\bar{0}\bar{0}\ldots\bar{0}\rangle + (-1)^k|\bar{1}\bar{1}\ldots\bar{1}\rangle$

$|\bar{0}\rangle \propto |+\rangle + |-\rangle \quad |\bar{1}\rangle \propto |+\rangle - |-\rangle$

Z basis

X basis

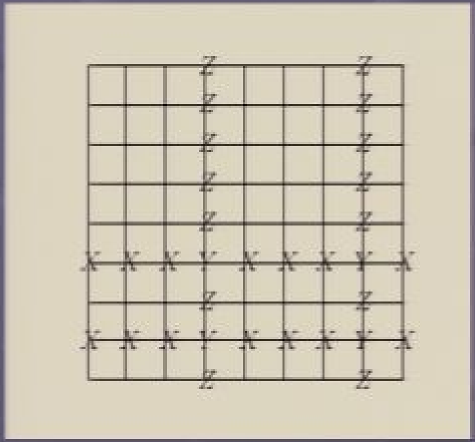
Bacon-Shor Codes

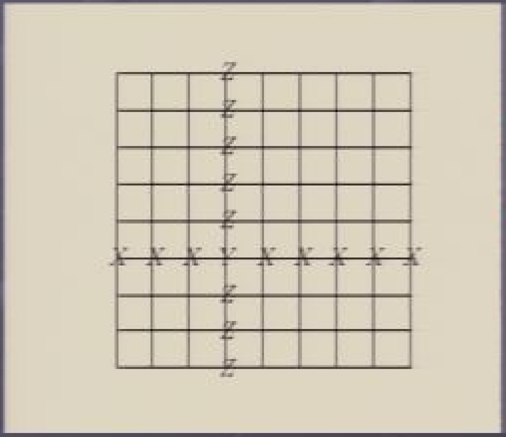
- **This code is a generalization of Shor's original quantum error correcting subspace code.**
- It eliminates the asymmetry inside Shor Code in the treatment of Z errors and X errors.
- This code is able to correct n/2 X and Z errors and it's generated by

$$S_{BS} = \left\langle X_{row-i,row-(i+1)}; Z_{col-j,col-(j+1)} : \boldsymbol{i}, \boldsymbol{j} \in \text{¢}_{n-1} \right\rangle$$

Stabilizer generators

Z in every couple of columns
X in every couple of rows





Logical qubits

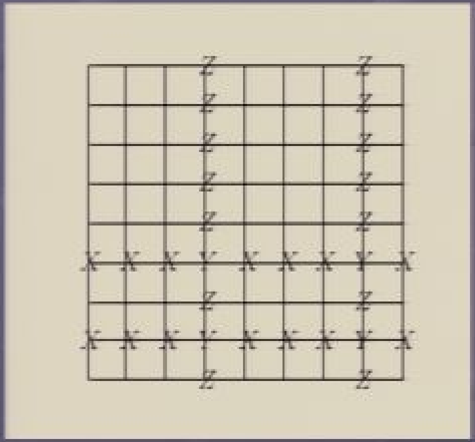
Bacon-Shor Codes

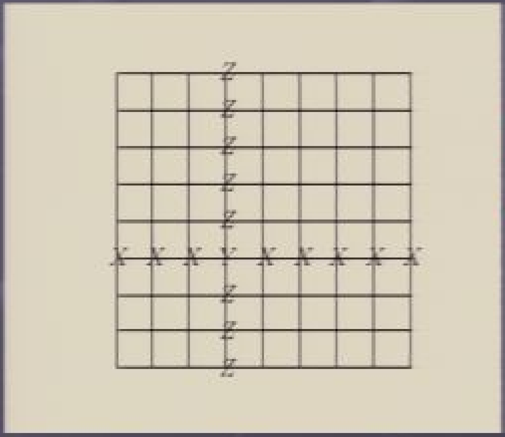
- **This code is a generalization of Shor's original quantum error correcting subspace code.**
- It eliminates the asymmetry inside Shor Code in the treatment of Z errors and X errors.
- This code is able to correct n/2 X and Z errors and it's generated by

$$S_{BS} = \left\langle X_{row-i,row-(i+1)}; Z_{col-j,col-(j+1)} : \boldsymbol{i}, \boldsymbol{j} \in \text{¢}_{n-1} \right\rangle$$

Stabilizer generators

Z in every couple of columns
X in every couple of rows





Logical qubits

Z in odd columns
X in odd rows

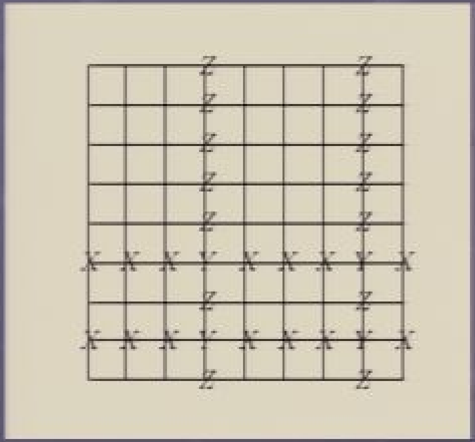
Bacon-Shor Codes

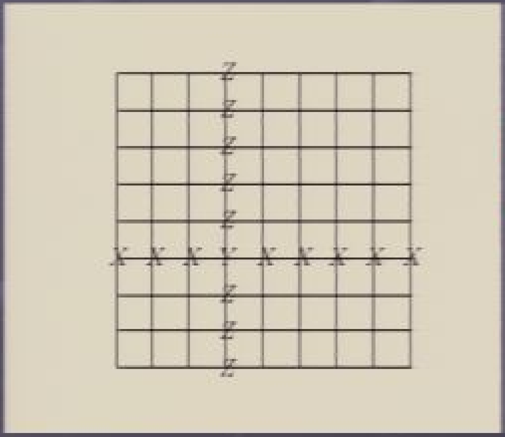
- **This code is a generalization of Shor's original quantum error correcting subspace code.**
- It eliminates the asymmetry inside Shor Code in the treatment of Z errors and X errors.
- This code is able to correct n/2 X and Z errors and it's generated by

Stabilizer generators

$$S_{BS} = \left\langle X_{row-i,row-(i+1)} ; Z_{col-j,col-(j+1)} : i, j \in \text{¢}_{n-1} \right\rangle$$

Z in every couple of columns
X in every couple of rows





Logical qubits

Z in odd columns
X in odd rows

Subsystem Constructio

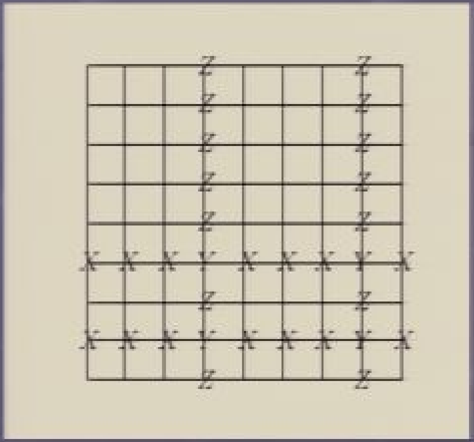
Bacon-Shor Codes

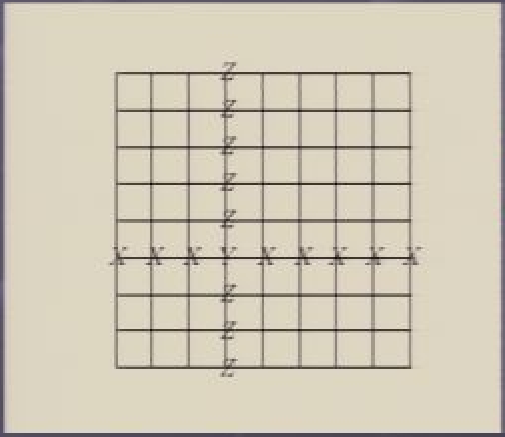
- **This code is a generalization of Shor's original quantum error correcting subspace code.**
- It eliminates the asymmetry inside Shor Code in the treatment of Z errors and X errors.
- This code is able to correct n/2 X and Z errors and it's generated by

$$S_{BS} = \left\langle X_{row-i,row-(i+1)}; Z_{col-j,col-(j+1)} : i, j \in \text{¢}_{n-1} \right\rangle$$

Stabilizer generators

Z in every couple of columns
X in every couple of rows





Logical qubits

Z in odd columns
X in odd rows

Subsystem Construction of this code allows simpler correcting routines and interesting fault tolerant properties!!!

Conclusions and future works

- Generalized construction of the Bacon Shor codes from two classical linear codes (Bacon- Casaccino, 2006)--> visit me during the poster session!
- Fault tolerant properties and considerable ancilla bit savings in the stabilizers measures (Aliferis-Cross, 2007)

Conclusions and future works

- Generalized construction of the Bacon Shor codes from two classical linear codes (Bacon- Casaccino, 2006)--> visit me during the poster session!
- Fault tolerant properties and considerable ancilla bit savings in the stabilizers measures (Aliferis-Cross, 2007)
- Generalization of the construction and remarkable properties about Singleton and Quantum Hamming Bound (Klappenecker, Sarvepalli, 2007)
- Work in progress…

Thank you for your attention!!!

A *Look* to the *future* . . .

Only time will tell if and when the problems of building a quantum computer can be overcome.As information becomes the worlds most valuable commodity, the economic, political and military fate of nations will depend on the strength of ciphers. Consequently, the development of a fully operational quantum computer would imperil our personal privacy, destroy electronic commerce and demolish the concept of national security. A quantum computer would jeopardize the stability of the world. Whichever country gets there first will have the ability to monitor the communications of its citizens, read the minds of its commercial rivals and eavesdrop on the plans of its enemies. Although it is still in its infancy, quantum computing presents a potential threat to the individual, to international business and to global security. -

Simon Singh

www.Simonsingh.net

Fault Tolerant Quantum computation

- Fault Tolerance computation requires that if the probability of introducing error in the circuit is p, the probability that the circuit brings two or more errors grows like $O(p^2)$ This means that a fault tolerant procedure comes to end succesfully with $1-cp^2$ probability where c depends only on the circuit

Subsystem Coding

- Recently it was realized that, since the most general method for encoding quantum information is to encode it into a subsystem, there exists a novel form of quantum error correction beyond the traditional quantum error correcting subspace codes
- These new quantum error correcting subsystem codes differ from subspace codes in that their quantum correcting routines can be considerably simpler than related subspace codes

Subsystem Space

$$H = \left(D \otimes C\right) \oplus \mathcal{E}$$

Code Space

$$H = \bigoplus_{s_1^X,\dots,s_{n-1}^X,s_1^Z,\dots,s_{n-1}^Z=\pm 1} H^T_{s^X,s^Z} \otimes H^L_{s^X,s^Z}$$