Lecture 16: Encoding of Stabilizer Codes

Stabilizer (group): $\mathcal{S} = \langle g_1, \ldots, g_{n-k} \rangle \subseteq \{id, \sigma_x, \sigma_z, \sigma_y\}^{\otimes n}$

Normalizer (group): $\mathcal{N} = \{ \pi \in \mathcal{P}_n : \pi \cdot s = s \cdot \pi \;\; \forall s \in \mathcal{S} \}$

as classical codes

$$\text{normalizer}\left\{\begin{array}{l}\left.\begin{array}{c} g_1\\ \vdots\\ g_{n-k}\end{array}\right\}\text{stabilizer}\\ \hline \left.\begin{array}{c}\bar{Z}_1\\ \vdots\\ \bar{Z}_k\end{array}\right\}\text{encoded }\sigma_z\text{-operations}\\ \left.\begin{array}{c}\bar{X}_1\\ \vdots\\ \bar{X}_k\end{array}\right\}\text{encoded }\sigma_x\text{-operations}\end{array}\right.$$

usually: stabilizer code is the joint $+1$-eigenspace of all operators in $\mathcal{S}$

more generally: for every <u>generator</u> g_i one can choose either the $+1$ or the -1 eigenspace to define the code

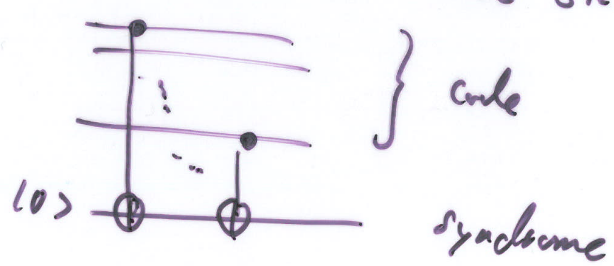
$\rightarrow$
- 2^{n-k} "equivalent versions" of the code
- decomposition of the full space into eigenspaces of $\mathcal{S}$

Any operator that changes the assignments of eigenvalues to the generators g_i can be detected by measuring the eigenvalue.
The sequence of $n-k$ eigenvalues yields a syndrome of the error.

16.1 Computing an error syndrome

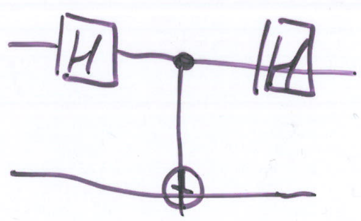
recall: CSS-codes

Compute the binary parity of the basis states using CNOTs

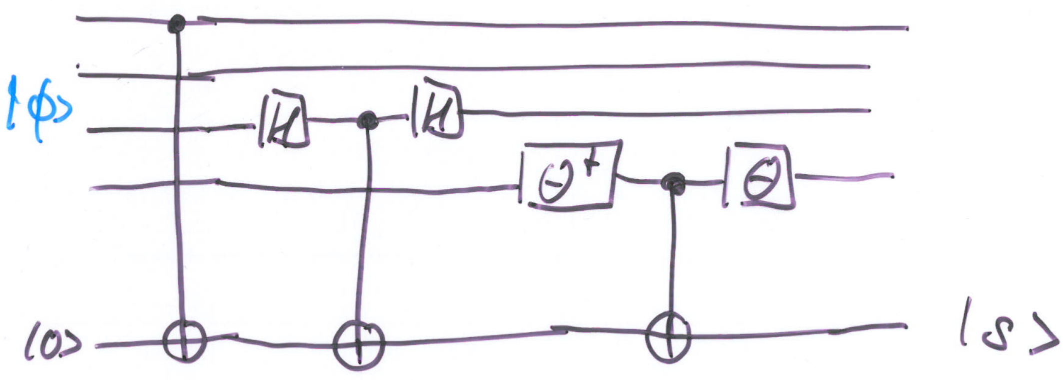


- CNOT "measures" of the eigenvalues of σ_z with eigenvector $|0\rangle$ and $|1\rangle$

$\hookrightarrow$ use local transformations to "measure" the eigenvalues of σ_x and σ_y, e.g. for σ_x:



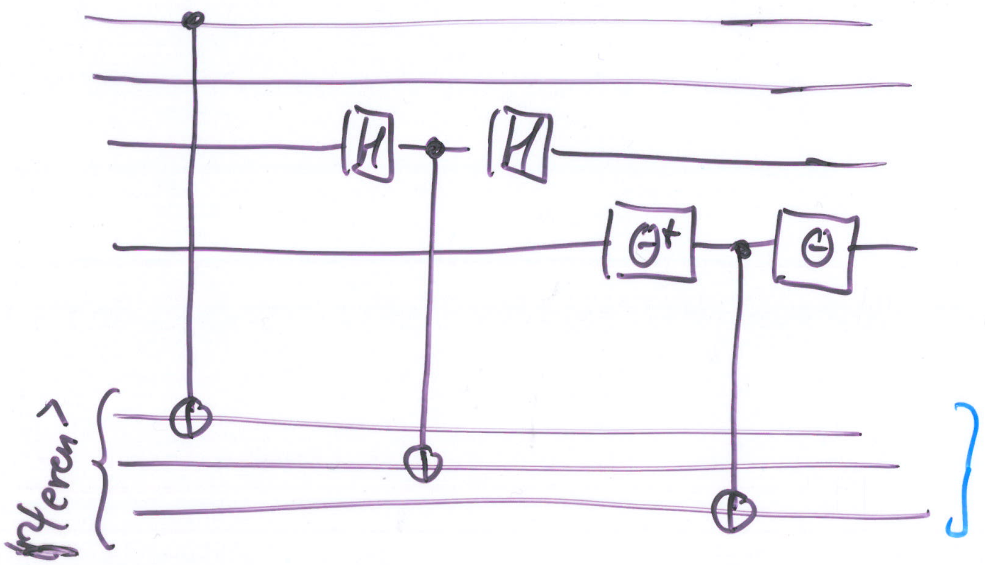
Example: $g = \sigma_z \otimes id \otimes \sigma_x \otimes \sigma_y$



$|s\rangle = |0\rangle$ if $g|\phi\rangle = |\phi\rangle$

$|s\rangle = |1\rangle$ if $g|\phi\rangle = -|\phi\rangle$

fault tolerant version



measure and compute the „classical" parity of the result

16.2 Encoding Circuits

Version of Cleve & Gottesman

first idea: the code is stabilized by any element of S, i.e.

$$\forall |\psi_i\rangle \in \mathcal{C}: \forall s \in \mathcal{S}: \quad s|\psi_i\rangle = |\psi_i\rangle$$

$$\hookrightarrow \quad \frac{1}{\sqrt{|S|}} \sum_{s \in \mathcal{S}} s|\phi_0\rangle =: |\psi_0\rangle$$

$$s'|\psi_0\rangle = \sum_{s \in \mathcal{S}} s's|\phi_0\rangle = \sum_{s \in \mathcal{S}} s|\phi_0\rangle$$

$\Rightarrow$ $|\psi_0\rangle$ is an element of the code (or the zero vector)

The operation $\frac{1}{|S|} \sum_{s \in \mathcal{S}} s$ is a projection onto the code space.

„Version 0" to realize a projection onto the code space:

- measure the eigenvalues $e_i \in \{+1, -1\}$ of the generators g_i
- the resulting state will be in the error space with error syndrome $(e_1, \ldots, e_{n-k})$.
- „correct" the corresponding error

Observation:

S is an abelian group generated by $g_1, \ldots, g_{n-k}$ with 2^{n-k} elements

$$\sum_{s \in S} s = \prod_{i=1}^{n-k} (I + g_i)$$

$$= (I + g_1)(I + g_2)(I + g_3) \cdot \ldots$$

$$= (I + g_1 + g_2 + g_1 \cdot g_2)(I + g_3) \cdot \ldots$$

remaining task:

Implement the projection operation $\frac{1}{2}(I + g_i)$ acting on some particular state by a unitary operation.

Case I: $\quad I + \sigma_x$

$$\frac{1}{2}(I + \sigma_x) = \frac{1}{2}\begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$$

$$\frac{1}{2}(I + \sigma_x)\,|0\rangle = \frac{1}{2}(|0\rangle + |1\rangle)$$

$$(H)\,|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$$

On the state $|0\rangle$, we can replace $I + \sigma_x$ by H

now $g_i = \sigma_x \otimes g_i'$

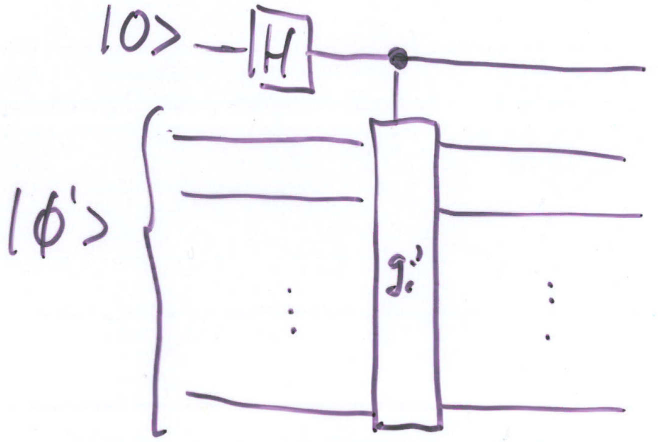
$$(I + g_i) = (I + \sigma_x \otimes g_i')$$

consider the state $|0\rangle \otimes |\phi'\rangle$

$$(I + g_i)(|0\rangle|\phi'\rangle) = |0\rangle|\phi'\rangle + |1\rangle(g_i'|\phi'\rangle)$$

$\rightarrow$ we can replace σ_x by H

and g_i' by $|0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes g_i'$



$$\hat{=} \quad (I + g_i')(|0\rangle|\phi'\rangle)$$

Case II: $g_i = \sigma_y \otimes g_i'$

$$(I + \sigma_y) = \begin{bmatrix} 1 & -i \\ i & 1 \end{bmatrix}$$

on $|0\rangle$: $|0\rangle - i|1\rangle$

use $P = \begin{bmatrix} 1 & \\ & i \end{bmatrix}$ and $H = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$

$\rightarrow$ replace $I + \sigma_y$ by $P^+ H$

Case III: $g_i = \sigma_z \otimes g_i'$

$$\frac{1}{2}(I + \sigma_z) = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$$

$\Longrightarrow$ the approach does not work for σ_z

$\Longrightarrow$ avoid σ_z in the construction!

additionally, we have to encode information!

- the state $|\psi_0\rangle$ corresponds to the encode state $|\underline{0}\ldots\underline{0}\rangle$ of k encoded qubits
- we can use the operations $\bar{X}_i$ to obtain an encode state $|\underline{a_1},\ldots,\underline{a_k}\rangle$:

$$\bar{X}_1^{a_1} \cdot \ldots \cdot \bar{X}_k^{a_k} |\underline{0}\ldots\underline{0}\rangle$$

$$= \bar{X}_1^{a_1} \cdot \ldots \cdot \bar{X}_k^{a_k} \Big(\sum_{s\in S} s\Big) |\phi_0\rangle$$

$$= \Big(\sum_{s\in S} s\Big) \bar{X}_1^{a_1} \cdot \ldots \cdot \bar{X}_k^{a_k} |\phi_0\rangle$$

$$= \prod_i (I + g_i) \Big(\prod_j \bar{X}_j^{a_j}\Big) |\phi_0\rangle$$

goal: implement the operation

$$\overline{X}_j^{a_j} | \ldots a_j \ldots \rangle$$

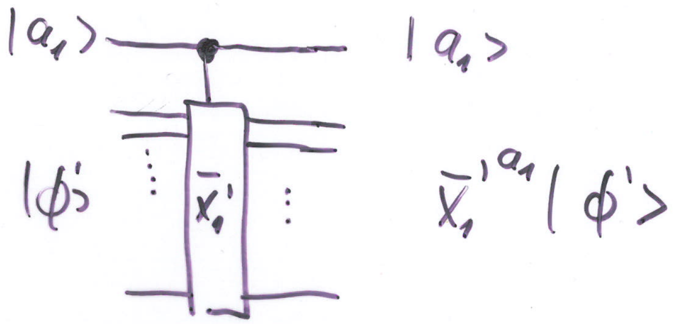
$j=1$: $$\overline{X}_1^{a_1} \left(| a_1 \rangle \otimes | \phi' \rangle \right)$$

Case I: $\overline{X}_1 = \sigma_x \otimes \overline{X}_1'$

Case II: $\overline{X}_1 = \sigma_y \otimes \overline{X}_1'$

two cases: $a_1 = 1$: apply $\overline{X}_1'$ to the rest

$a_1 = 0$: do nothing



$\hookrightarrow$ we have realized

$$\overline{X}^{a_1} |0\rangle \ldots |0\rangle$$

$\Rightarrow$ "pre-encoded state" $\overline{X}_1^{a_1} \ldots \overline{X}_k^{a_k} \left(| \underbrace{0 \ldots 0}_{k} \rangle | \underbrace{0 \ldots 0}_{n-k} \rangle \right)$

is replaced by an operation on the input state

$$| a_1 \ldots a_k \rangle | 0 \ldots 0 \rangle$$

We have to ensure that we always find a position in our generators g_i and encoded operation $\overline{X}_j$ such that

i) the Pauli matrix at this position is either σ_x or σ_y

ii) the state at this position is $|0\rangle$

$\Rightarrow$ compute a transformed „stabilizer + encoded X" matrix $\left[\begin{array}{c} S' \\ \hline X' \end{array}\right]$ in „upper triangular" form

idea: use row and ~~column~~ operations on the stabilizer + encoded X

$$G = \begin{bmatrix} 1 & 1 & \omega & 0 & \omega \\ 1+\omega & 0 & 1+\omega & \omega & \omega \\ 1+\omega & \omega & \omega & 1+\omega & 0 \\ 1 & \omega & 0 & \omega & 1 \end{bmatrix} = \begin{bmatrix} X & X & Z & I & Z \\ Y & I & Y & Z & Z \\ Y & Z & Z & \cancel{Z}\,Y & I \\ X & Z & I & Z & X \\ \hline a & b & \cancel{c} & \cancel{d} & \cancel{e} \\ X & I & Z & Z & I \end{bmatrix}$$

same ~~encode~~ operation

choose the operation $\overline{X}_1$ after transforming the stabilizer in „triangular form"

$$G' = \begin{bmatrix} 1 & 1 & \omega & 0 & \omega \\ \omega^2 & 0 & 1+\omega & \omega & \omega \\ \omega^2 & \omega & \omega & 1+\omega & 0 \\ 1 & \omega & 0 & \omega & 1 \end{bmatrix} = \begin{array}{c} \begin{array}{ccccc} X & Z^a & Z^b & Z^d & Z^e \end{array} \\ \begin{bmatrix} X & \boxed{X} & Z & I & Z \\ Y & I & \boxed{Y} & Z & Z \\ Y & Z & Z & \boxed{Y} & I \\ X & Z & I & Z & \boxed{X} \end{bmatrix} \\ \begin{bmatrix} X & I & Z & Z & I \end{bmatrix} \end{array}$$

encoded $\bar{X}_1$

without proof:

we can achieve a form of the stabilizer such that

- above the „diagonal" there are only I or Z
- on the diagonal, there are only X or Y

use the local Clifford operations H and Θ and row additions, i.e multiplications of generators