Lecture 12: Circuits for CSS codes

12.1 Computing the error syndrome

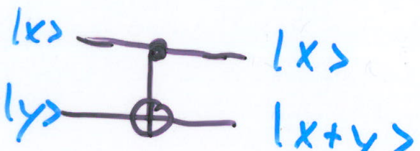
$$C_2^\perp \le C_1 = [n, k_1, d_1] \qquad C_1^\perp \le C_2 = [n, k_2, d_2]$$

$$G_1 = \underbrace{\begin{bmatrix} H_2 \\ \hline D_1 \end{bmatrix}}_{n} \begin{matrix} n-k_2 \end{matrix} \Big\} k_1 \qquad G_2 = \underbrace{\begin{bmatrix} H_1 \\ \hline D_2 \end{bmatrix}}_{n} \begin{matrix} n-k_1 \end{matrix} \Big\} k_2$$

syndrome for bit-flip (σ_x-) errors:

$$S_x: \quad \underbrace{|x\rangle}_{n} \underbrace{|y\rangle}_{n-k_1} \mapsto |x\rangle \, |x \cdot H_1^t + y\rangle$$

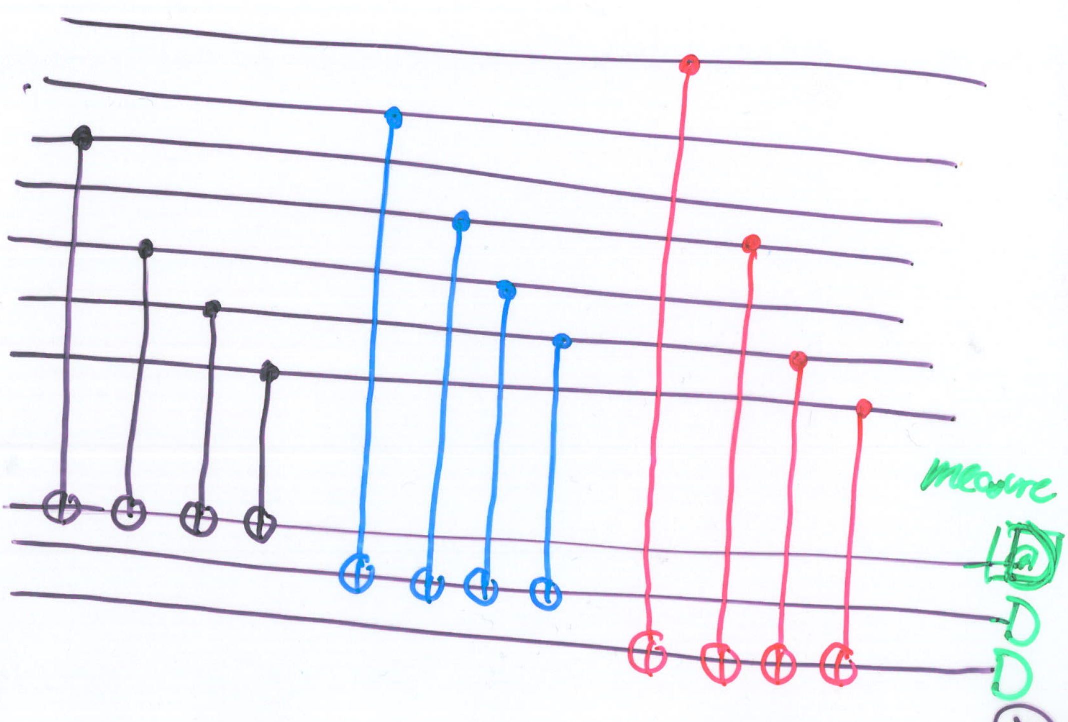
CNOT:



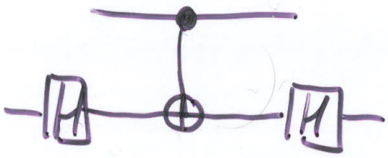
$$\begin{pmatrix} 1 & & & \\ & 1 & & \\ & & 0 & 1 \\ & & 1 & 0 \end{pmatrix}$$

$$H_1^t = \begin{array}{c} \begin{bmatrix} & & 1 \\ & 1 & \\ 1 & & \\ 0 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix} 7 \\ 3 \end{array}$$

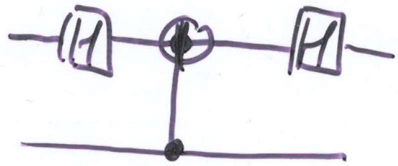
$\to 3$



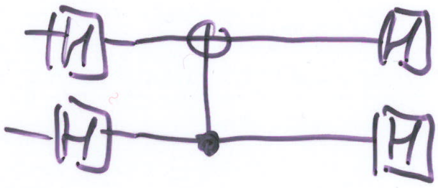
problem: This way of computing an error syndrome is not fault tolerant, as many control qubits are coupled to the same target qubit $\Rightarrow$ single errors may spread over many locations



$$\hat{=} \begin{pmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & -1 \end{pmatrix} =$$

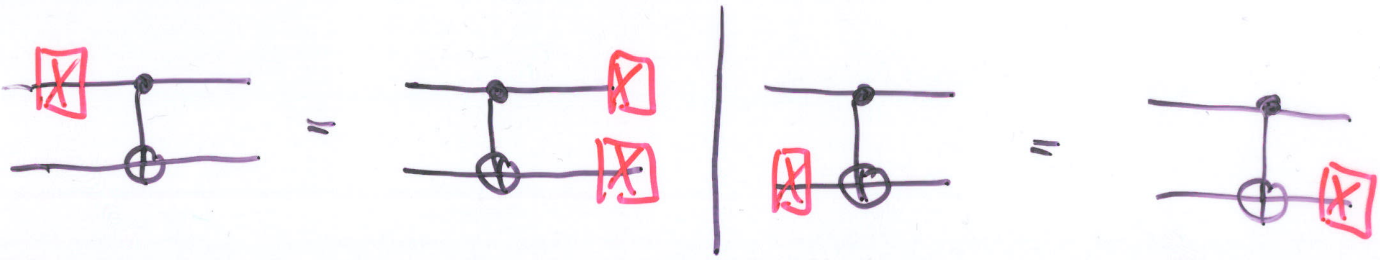


$=$

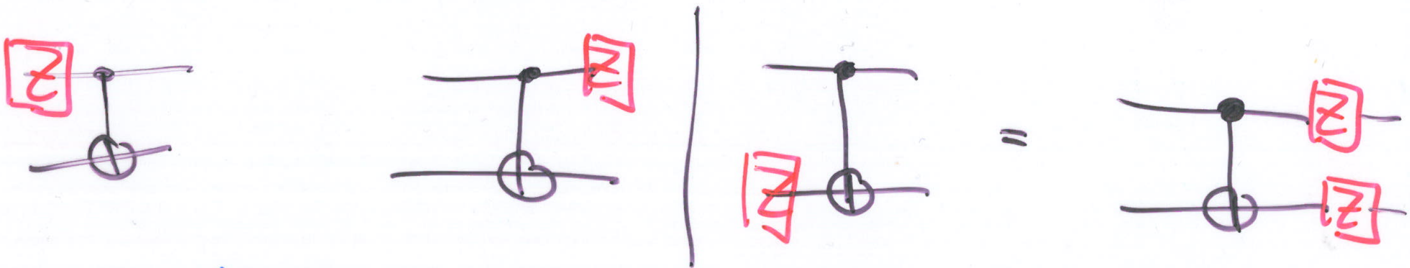


error propagation for CNOT:

X:

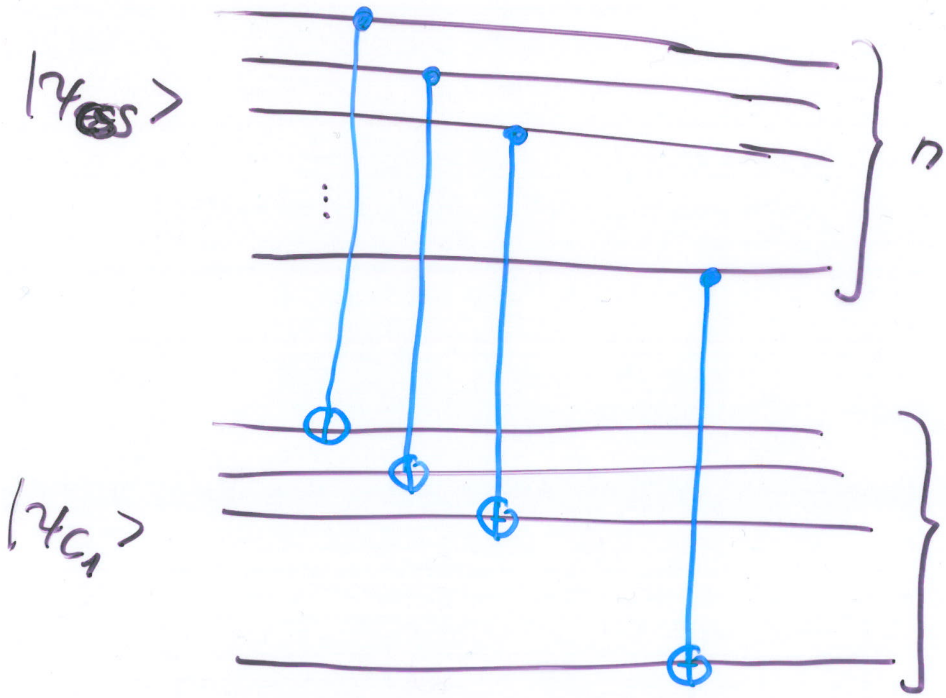


Z:



X propagates from control to target
Z " " target to control

12.2 Fault tolerant syndrome computation



„transversal CNOT"

$$|\psi_{CSS}\rangle = \sum_{c\in C_1} \beta_c |c\rangle$$

$$|\psi_{C_1}\rangle = \sum_{c'\in C_1} |c'\rangle$$

$$\left(CNOT^{\otimes n}\right) |\psi_{CSS}\rangle |\psi_{C_1}\rangle$$

$$= \sum_{e_x} \gamma_{e_x} \sum_{c\in C_1} \beta_c |c + e_x\rangle \sum_{c'\in C_1} |c + c' + e_x\rangle$$

$\downarrow$ measurement

random codeword $\tilde{c} \in C_1$

or $\tilde{c} + \tilde{e}_x$

then compute $(\tilde{c} + \tilde{e}_x)\cdot H_1^t = \tilde{e}_x \cdot H_1^t$

on a classical computer