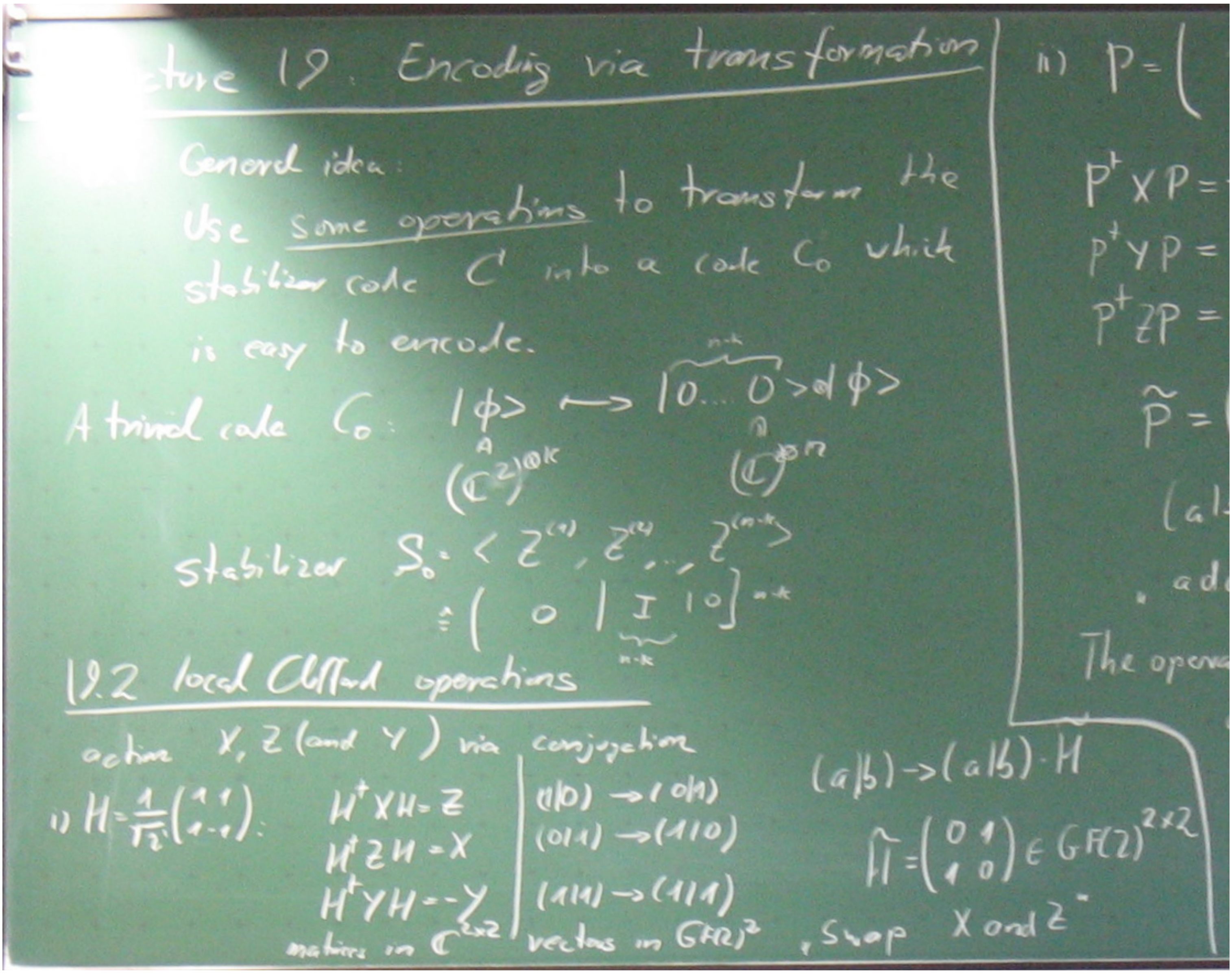
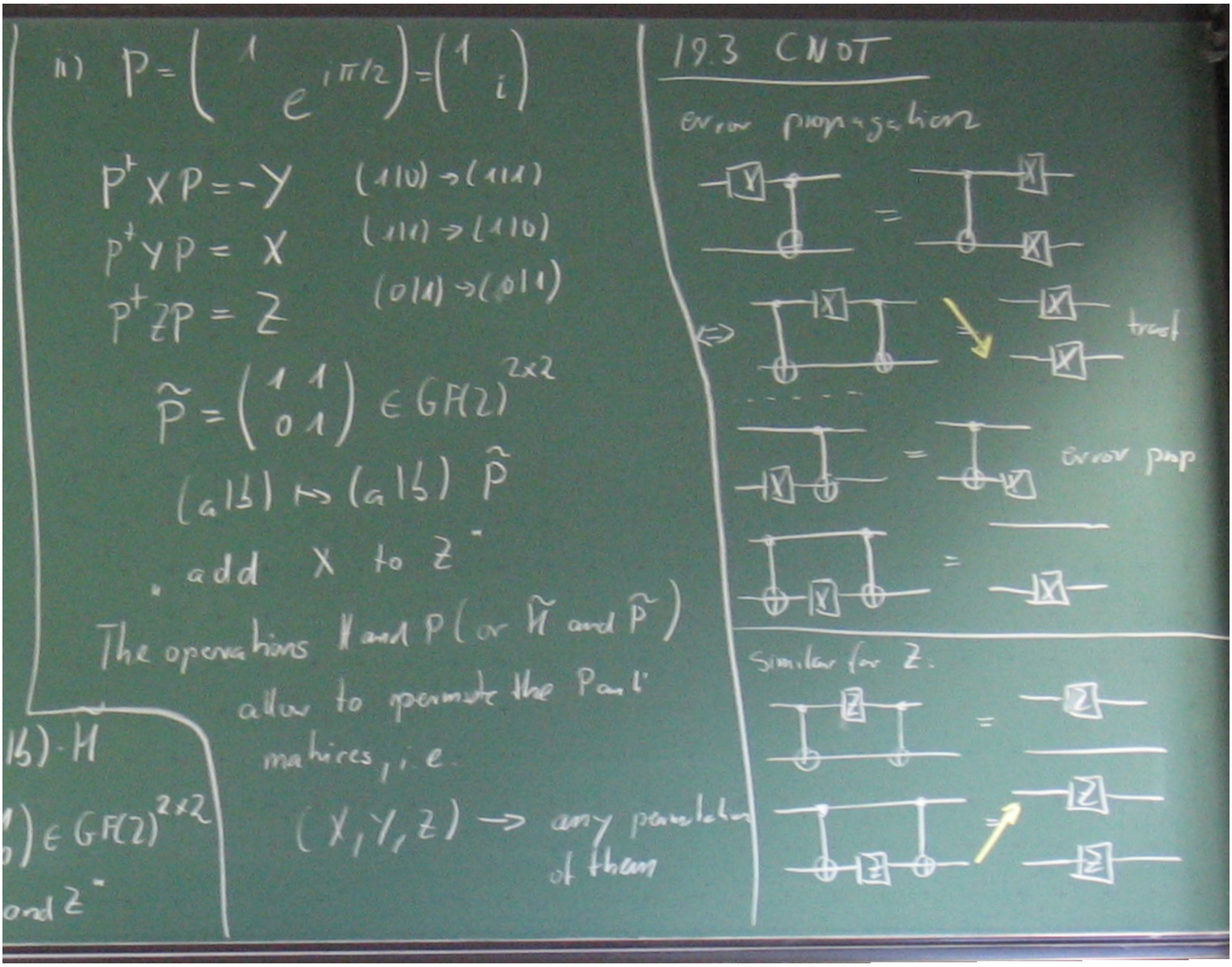
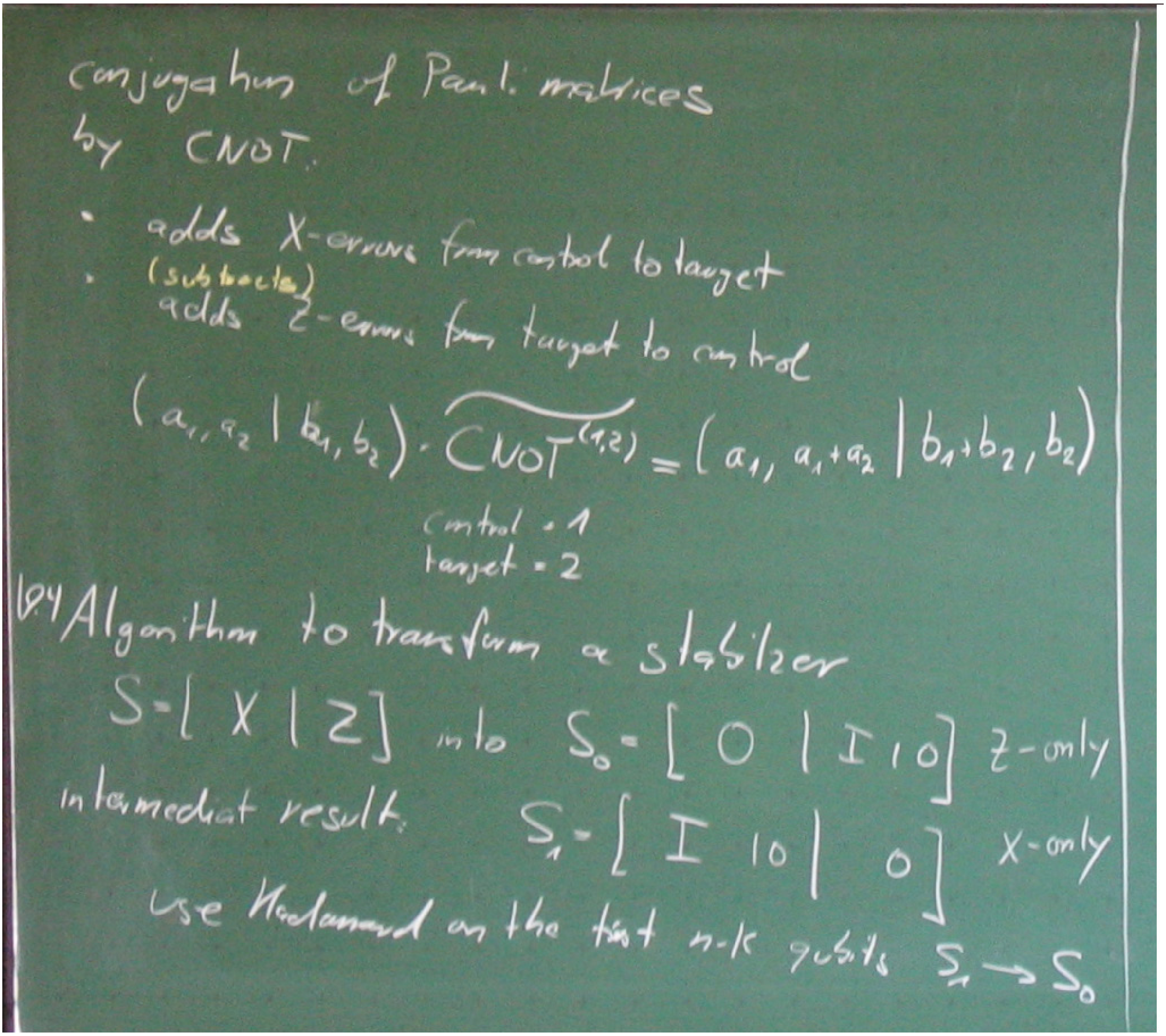
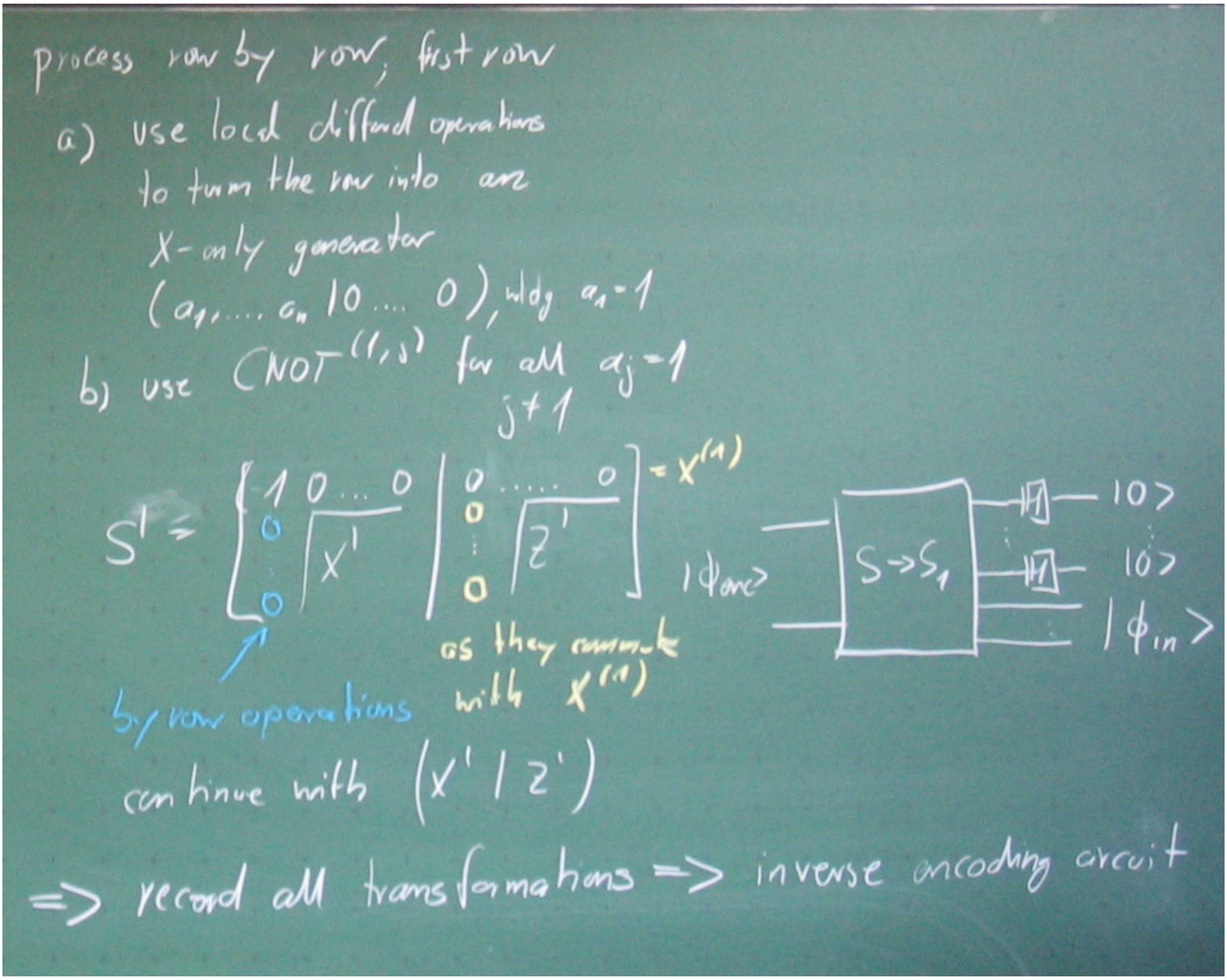

Encoding via transformation
General idea:
Use some operations to transform the stabilizer code C into a code C0 which is easy to encode.
A trivial code C0:
stabilizer
19.2 local Clifford operations
action X, Z (and Y) via conjugation
H+XH = Z
H+ZH = X
H+YH = -Y
matrices in C2x2
vectors in GF(2)2
(a|b) -> (a|b)·H
∈ GF(2)2x2
swap X and Z
P+XP =
P+YP =
P+ZP =
The opera


ii) P = (1 e^{iπ/2}) = (1 i)
P⁺XP = −Y (1|0) → (1|1)
P⁺YP = X (1|1) → (1|0)
P⁺ZP = Z (0|1) → (0|1)
P̃ = (1 1 / 0 1) ∈ GF(2)^{2×2}
(a|b) ↦ (a|b) P̃
"add X to Z"
The operations H and P (or H̃ and P̃) allow to permute the Pauli matrices, i.e.
(X, Y, Z) → any permutation of them
19.3 CNOT
error propagation
similar for Z.
error prop
and Z"


conjugation of Pauli matrices
by CNOT:
• adds X-errors from control to target
• (subtracts)
adds Z-errors from target to control
(a1, a2 | b1, b2) · CNOT^(1,2) = (a1, a1+a2 | b1+b2, b2)
control = 1
target = 2
8.4 Algorithm to transform a stabilizer
S = [X | Z] into S0 = [0 | I | 0] Z-only
intermediat result: S1 = [I | 0 | 0] X-only
use Hadamard on the first n-k qubits S1 → S0


process row by row; first row
a) use local clifford operations
to turn the row into an
X-only generator
(a1, ... an | 0 .... 0), wlog a1=1
b) use CNOT(1,j) for all aj=1
j≠1
S' = [1 0 ... 0 | 0 ..... 0] = X(1)
X'
Z'
|φenc>
S->S1
|0>
|0>
|φin>
by row operations
as they commute
with X(1)
continue with (X' | Z')
=> record all transformations => inverse encoding circuit

References

- Markus Grassl,
  "Encoding and decoding quantum error-correcting codes,,
  in: G. Casati, D. L. Shepelyansky, and P. Zoller,
  Proceedings of the International School of Physics "E. Fermi",
  Course CLXII on "Quantum Computers, Algorithms and Chaos",
  IOS Press, 2006, pp. 33–44.
- Markus Grassl,
  "Quantum Error Correction,"
  International School of Physics "Enrico Fermi,"
  Course CLXII on Quantum Computers, Algorithms and Chaos,
  Varenna, 12.07.2005.
  Transparencies of the talk.
- Markus Grassl, Martin Rötteler, and Thomas Beth,
  "Efficient Quantum Circuits for Non-Qubit Quantum Error-Correcting Codes,"
  International Journal of Foundations of Computer Science (IJFCS), Vol. 14, No. 5 (2003), pp. 757–775.
  Preprint quant-ph/0211014,

International School of Physics “Enrico Fermi”

Quantum Computers, Algorithms and Choas

Varenna, 5–15 July 2005

Quantum Error Correction

Markus Grassl

Arbeitsgruppe *Quantum Computing*, Prof. Beth
Institut für Algorithmen und Kognitive Systeme
Fakultät für Informatik, Universität Karlsruhe
Germany
http://iaks-www.ira.uka.de/QIV

Outline

- general setting of quantum error correction
- discretizing errors
- encoding circuits

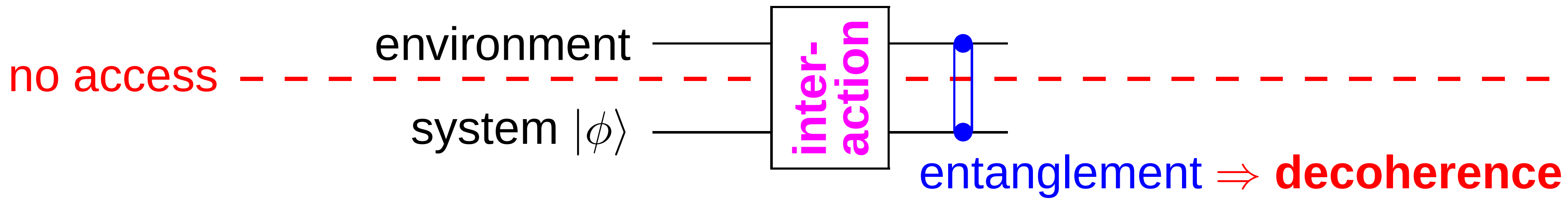
References

Markus Grassl, “Algorithmic aspects of quantum error-correcting codes”, in: Ranee K. Brylinski and Goong Chen (Eds.), *Mathematics of Quantum Computation*, Chapman & Hall/CRC, 2002, pp. 223-252.

Markus Grassl, Martin Rötteler, and Thomas Beth, “Efficient Quantum Circuits for Non-Qubit Quantum Error-Correcting Codes”, *International Journal of Foundations of Computer Science (IJFCS)*, Vol. 14, No. 5 (2003), pp. 757-775. Preprint quant-ph/0211014.

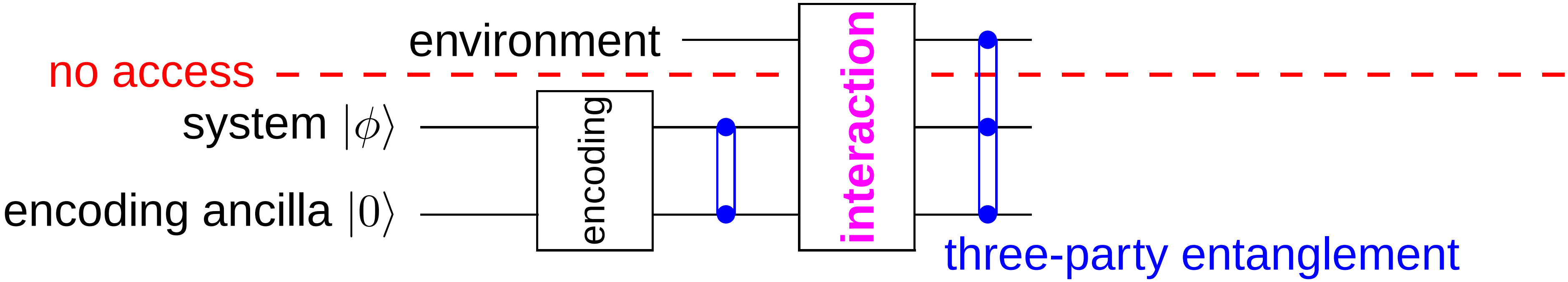
Quantum Error Correction

General scheme



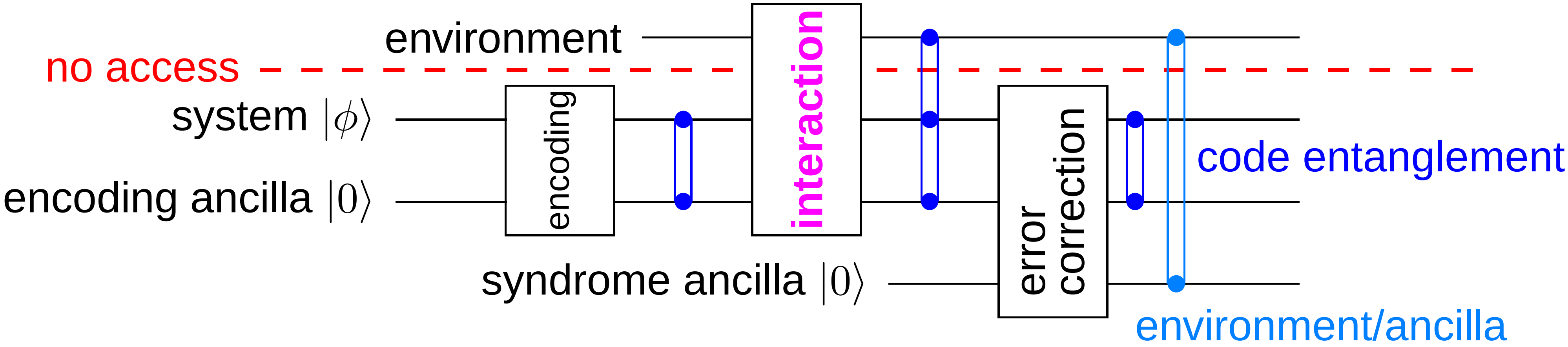
Quantum Error Correction

General scheme



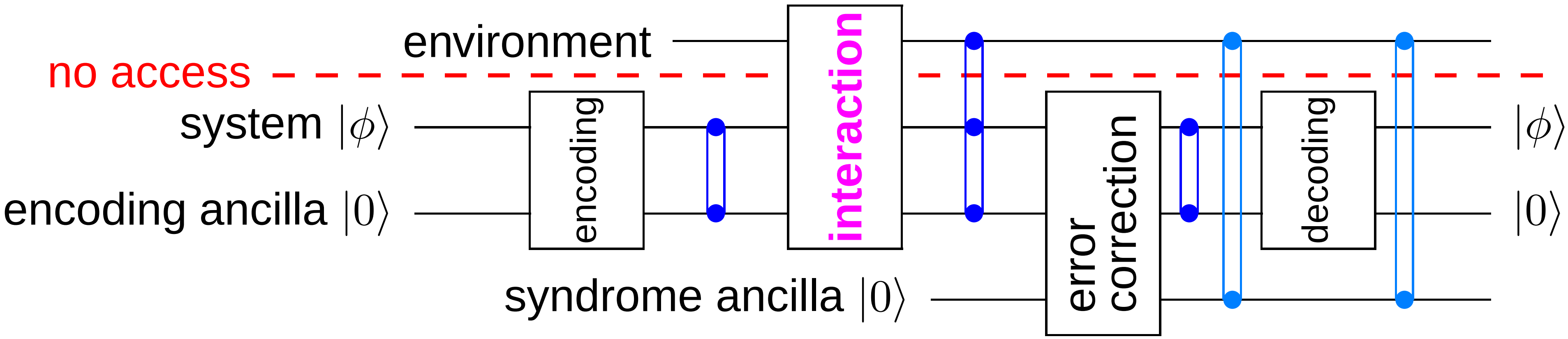
Quantum Error Correction

General scheme



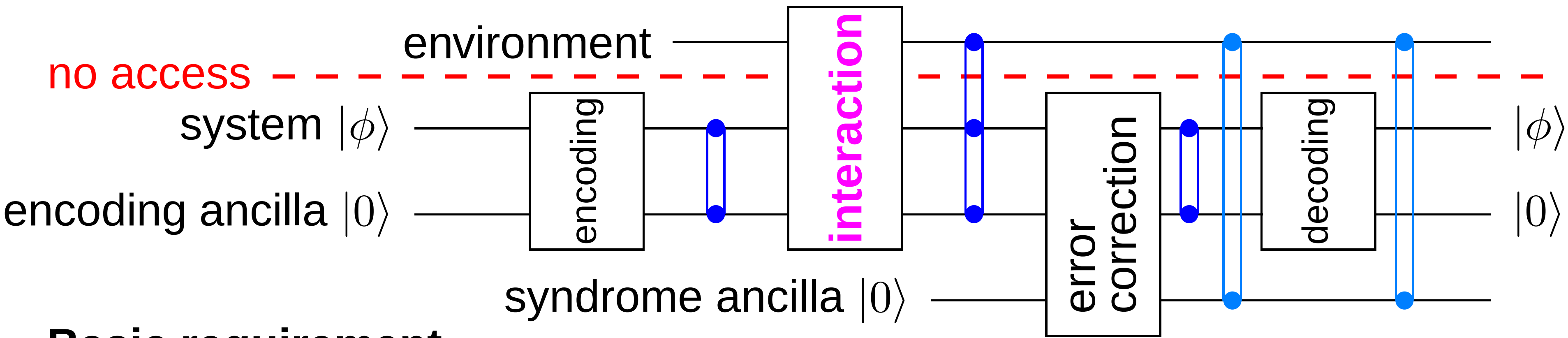
Quantum Error Correction

General scheme



Quantum Error Correction

General scheme



Basic requirement

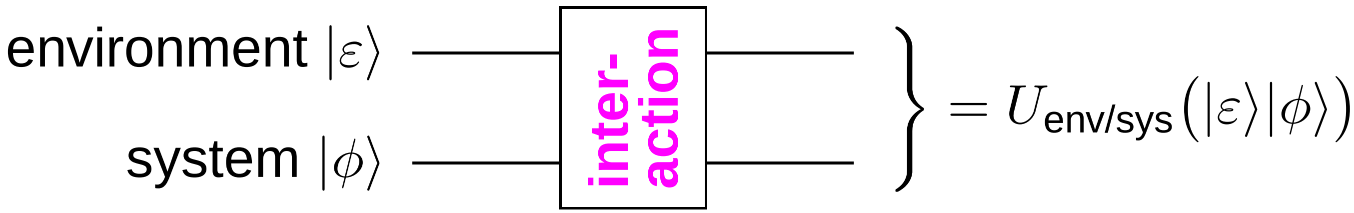
some knowledge about the interaction between system and environment

Common assumptions

- no initial entanglement between system and environment
- local or uncorrelated errors, i. e., only a few qubits are disturbed
  $\Longrightarrow$ CSS codes, stabilizer codes
- interaction with symmetry $\Longrightarrow$ decoherence free subspaces

Interaction System/Environment

“Closed” System



“Channel”

$$\mathsf{Q}\colon \rho_{\text{in}} := |\phi\rangle\langle\phi| \longmapsto \rho_{\text{out}} := \mathsf{Q}(|\phi\rangle\langle\phi|) := \sum_i E_i \rho_{\text{in}} E_i^\dagger$$

with Kraus operators (error operators) E_i

Local/low correlated errors

- product channel $\mathsf{Q}^{\otimes n}$ where Q is “close” to identity
- Q can be expressed (approximated) with error operators $\tilde{E}_i$ such that each E_i acts on few subsystems, e. g. quantum gates

Computer Science Approach: *Discretize*

QECC Characterization

[Knill & Laflamme, PRA **55**, 900–911 (1997)]

A subspace $\mathcal{C}$ of $\mathcal{H}$ with orthonormal basis $\{|c_1\rangle, \ldots, |c_K\rangle\}$ is an error-correcting code for the error operators $\mathcal{E} = \{E_1, E_2, \ldots\}$, if there exists constants $\alpha_{k,l} \in \mathbb{C}$ such that for all $|c_i\rangle$, $|c_j\rangle$ and for all $E_k, E_l \in \mathcal{E}$:

$$\langle c_i|E_k^\dagger E_l|c_j\rangle = \delta_{i,j}\alpha_{k,l}. \tag{1}$$

It is sufficient that (1) holds for a vector space basis of $\mathcal{E}$.

Linearity of Conditions

Assume that $\mathcal{C}$ can correct the errors $\mathcal{E} = \{E_1, E_2, \ldots\}$.

New error-operators:

$$A := \sum_k \lambda_k E_k \quad \text{and} \quad B := \sum_l \mu_l E_l$$

$$\begin{aligned} \langle c_i | A^\dagger B | c_j \rangle &= \sum_{k,l} \overline{\lambda_k} \mu_l \langle c_i | E_k^\dagger E_l | c_j \rangle \\ &= \sum_{k,l} \overline{\lambda_k} \mu_l \delta_{i,j} \alpha_{k,l} \\ &= \delta_{i,j} \cdot \alpha'(A, B) \end{aligned}$$

It is sufficient to correct error operators that form a basis of the linear vector space spanned by the operators $\mathcal{E}$.

$\Longrightarrow$ only a finite set of errors.

Error Basis

Pauli Matrices

$$\sigma_x := \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad \sigma_y := \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad \sigma_z := \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \quad I := \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

- vector space basis of all 2×2 matrices
- unitary matrices which generate a *finite* group

Error Basis for many Qubits/Qudits

$\mathcal{E}$ error basis for subsystem of dimension d with $I \in \mathcal{E}$
$\Longrightarrow \mathcal{E}^{\otimes n}$ error basis with elements

$$E := E_1 \otimes \ldots \otimes E_n, \quad E_i \in \mathcal{E}$$

weight of E: number of factors $E_i \neq I$

Local Error Model

Code Parameters

$\mathcal{C} = [\![n, k, d]\!]$

n: number of subsystems used in total

k: number of (logical) subsystems encoded

d: "minimum distance"

– correct all errors acting on at most $(d-1)/2$ subsystems

– detect all errors acting on less than d subsystems

– correct all errors acting on less than d subsystems at *known* positions

General Decoding Algorithm

	$E_1\mathcal{C}$	$E_2\mathcal{C}$	$\cdots$	$E_k\mathcal{C}$	$\cdots$
$\mathcal{V}_0$	$E_1\|c_0\rangle$	$E_2\|c_0\rangle$	$\cdots$	$E_k\|c_0\rangle$	$\cdots$
$\mathcal{V}_1$	$E_1\|c_1\rangle$	$E_2\|c_1\rangle$	$\cdots$	$E_k\|c_1\rangle$	$\cdots$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	
$\mathcal{V}_i$	$E_1\|c_i\rangle$	$E_2\|c_i\rangle$	$\cdots$	$E_k\|c_i\rangle$	$\cdots$
$\vdots$	$\vdots$	$\vdots$		$\vdots$	$\ddots$

$$\langle c_i|E_k^\dagger E_l|c_j\rangle = \delta_{i,j}\alpha_{k,l} \tag{1}$$

General Decoding Algorithm

	$E_1\mathcal{C}$	$E_2\mathcal{C}$	$\cdots$	$E_k\mathcal{C}$	$\cdots$
$\mathcal{V}_0$	$E_1\|c_0\rangle$	$E_2\|c_0\rangle$	$\cdots$	$E_k\|c_0\rangle$	$\cdots$
$\mathcal{V}_1$	$E_1\|c_1\rangle$	$E_2\|c_1\rangle$	$\cdots$	$E_k\|c_1\rangle$	$\cdots$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	
$\mathcal{V}_i$	$E_1\|c_i\rangle$	$E_2\|c_i\rangle$	$\cdots$	$E_k\|c_i\rangle$	$\cdots$
$\vdots$	$\vdots$	$\vdots$		$\vdots$	$\ddots$

rows are orthogonal as $\langle c_i|E_k^\dagger E_l|c_j\rangle = 0$ for $i \neq j$

$$\langle c_i|E_k^\dagger E_l|c_j\rangle = \delta_{i,j}\alpha_{k,l} \tag{1}$$

General Decoding Algorithm

	$E_1\mathcal{C}$	$E_2\mathcal{C}$	$\cdots$	$E_k\mathcal{C}$	$\cdots$
$\mathcal{V}_0$	$E_1\|c_0\rangle$	$E_2\|c_0\rangle$	$\cdots$	$E_k\|c_0\rangle$	$\cdots$
$\mathcal{V}_1$	$E_1\|c_1\rangle$	$E_2\|c_1\rangle$	$\cdots$	$E_k\|c_1\rangle$	$\cdots$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	
$\mathcal{V}_i$	$E_1\|c_i\rangle$	$E_2\|c_i\rangle$	$\cdots$	$E_k\|c_i\rangle$	$\cdots$
$\vdots$	$\vdots$	$\vdots$		$\vdots$	$\ddots$

rows are orthogonal as $\langle c_i|E_k^\dagger E_l|c_j\rangle = 0$ for $i \neq j$

inner product between columns is constant as $\langle c_i|E_k^\dagger E_l|c_i\rangle = \alpha_{k,l}$

$$\langle c_i|E_k^\dagger E_l|c_j\rangle = \delta_{i,j}\alpha_{k,l} \qquad (1)$$

General Decoding Algorithm

	$E_1\mathcal{C}$	$E_2\mathcal{C}$	$\cdots$	$E_k\mathcal{C}$	$\cdots$
$\mathcal{V}_0$	$E_1\|c_0\rangle$	$E_2\|c_0\rangle$	$\cdots$	$E_k\|c_0\rangle$	$\cdots$
$\mathcal{V}_1$	$E_1\|c_1\rangle$	$E_2\|c_1\rangle$	$\cdots$	$E_k\|c_1\rangle$	$\cdots$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	
$\mathcal{V}_i$	$E_1\|c_i\rangle$	$E_2\|c_i\rangle$	$\cdots$	$E_k\|c_i\rangle$	$\cdots$
$\vdots$	$\vdots$	$\vdots$		$\vdots$	$\ddots$

rows are orthogonal as $\langle c_i|E_k^\dagger E_l|c_j\rangle = 0$ for $i \neq j$

inner product between columns is constant as $\langle c_i|E_k^\dagger E_l|c_i\rangle = \alpha_{k,l}$

$\Longrightarrow$ simultaneous Gram-Schmidt orthogonalization within the spaces $\mathcal{V}_i$

Orthogonal Decomposition

	$E'_1\mathcal{C}$	$E'_2\mathcal{C}$	$\cdots$	$E'_k\mathcal{C}$	$\cdots$
$\mathcal{V}_0$	$E'_1\|c_0\rangle$	$E'_2\|c_0\rangle$	$\cdots$	$E'_k\|c_0\rangle$	$\cdots$
$\mathcal{V}_1$	$E'_1\|c_1\rangle$	$E'_2\|c_1\rangle$	$\cdots$	$E'_k\|c_1\rangle$	$\cdots$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	
$\mathcal{V}_i$	$E'_1\|c_i\rangle$	$E'_2\|c_i\rangle$	$\cdots$	$E'_k\|c_i\rangle$	$\cdots$
$\vdots$	$\vdots$	$\vdots$		$\vdots$	$\ddots$

rows are mutually orthogonal

columns are mutually orthogonal

- new error operators E'_k are linear combinations of the E_l
- projection onto $E'_k\mathcal{C}$ determines the error
- exponentially many orthogonal spaces $E'_k\mathcal{C}$

Encoding Stabilizer Codes

[Grassl, Rötteler, and Beth, IJFCS, **14** (2003), pp. 757-775]

Basic idea:

Use operations of the *generalized Clifford group* (or Jacobi group) to transform the stabilizer S into a trivial stabilizer $S_0 := \left\langle Z^{(1)}, \ldots Z^{(n-k)} \right\rangle$.

- row/column operations on the binary matrix $(X|Z)$ to obtain "normal form" $(0|I0)$
- operations on $(X|Z)$ correspond to
  - "elementary" single-qubit gates
  - CNOT-gate
  - single qudit gate $P := \begin{pmatrix} 1 & 0 \\ 0 & e^{\pi i/2} \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix} \in \mathbb{C}^{2\times 2}$

Action on Pauli Matrices

Hadamard matrix H	$HXH = Z$ $(1,0) \mapsto (0,1)$	$HYH = -Y$ $(1,1) \mapsto (1,1)$	$HZH = X$ $(0,1) \mapsto (1,0)$
	exchange X and Z		
matrix P	$P^\dagger XP = -Y$ $(1,0) \mapsto (1,1)$	$P^\dagger YP = X$ $(1,1) \mapsto (1,0)$	$P^\dagger ZP = Z$ $(0,1) \mapsto (0,1)$
	multiply X by Z		

in $\mathbb{C}$

$\bmod 2$

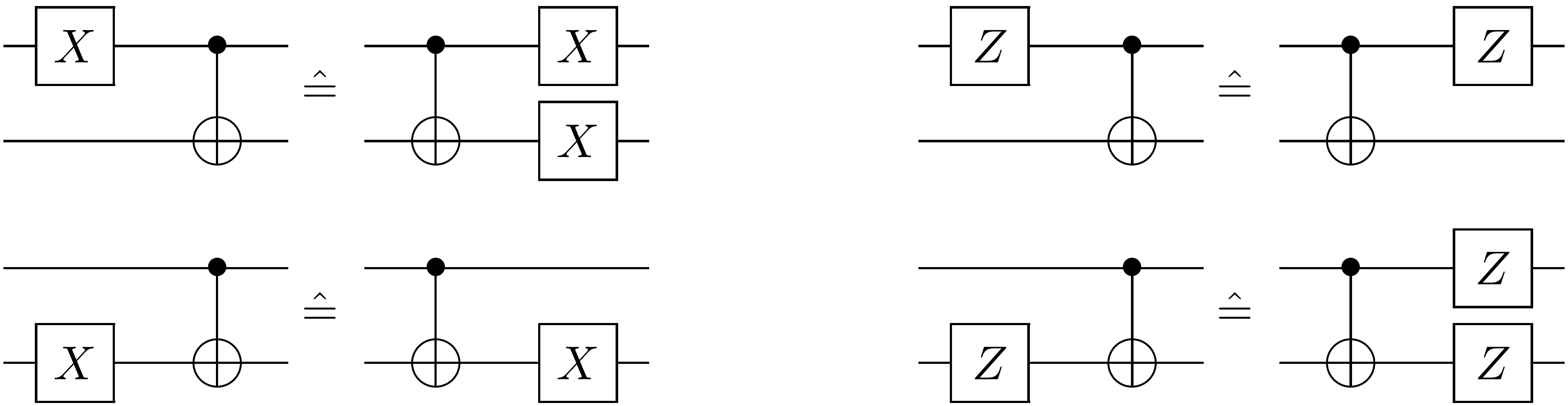
operation on binary row vectors: $(a,b)\widetilde{M} = (a',b')$ (arithmetic $\bmod 2$)

$$\widetilde{H} \mathrel{\hat{=}} \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \quad \text{and} \quad \widetilde{P} \mathrel{\hat{=}} \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}$$

local operation on $(X|Z)$:
multiplying column i in submatrix X and column i in submatrix Z by $\widetilde{M}$

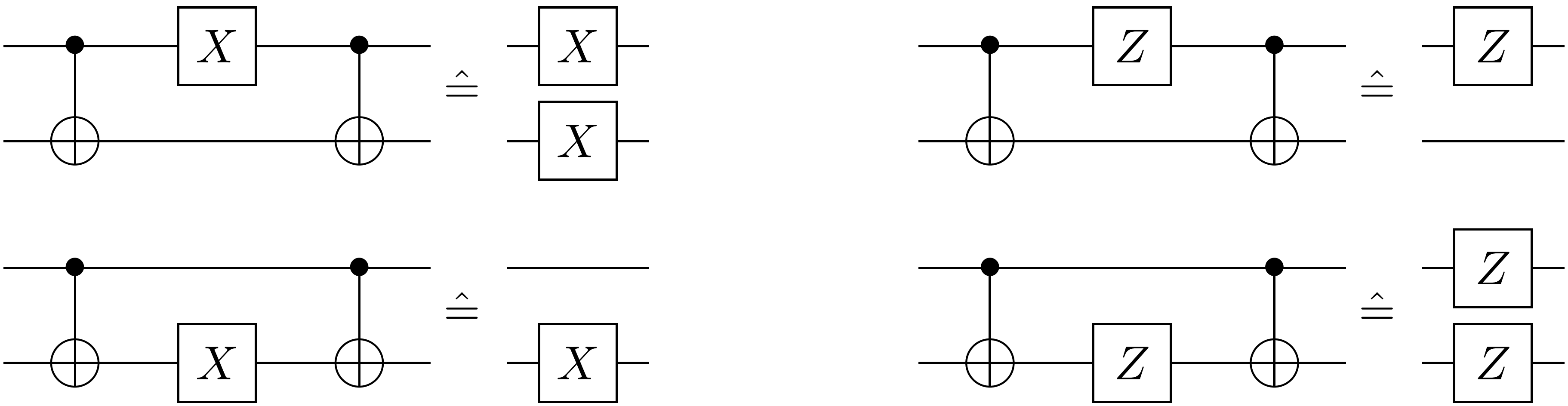
Action of CNOT

Error propagation



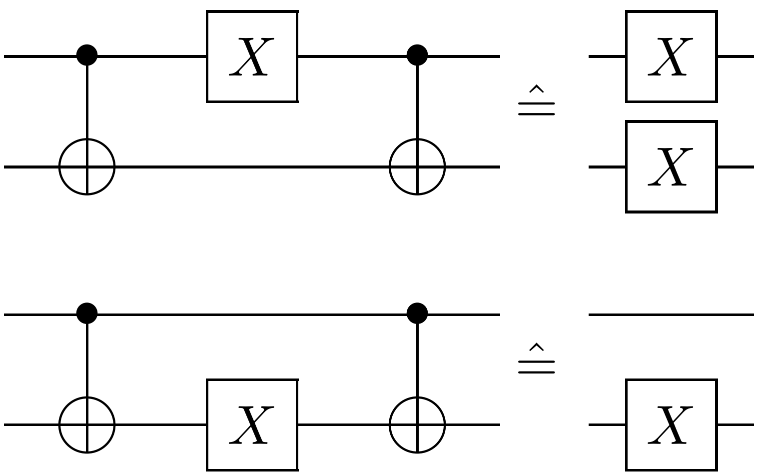
Action of CNOT

Modifying stabilizers

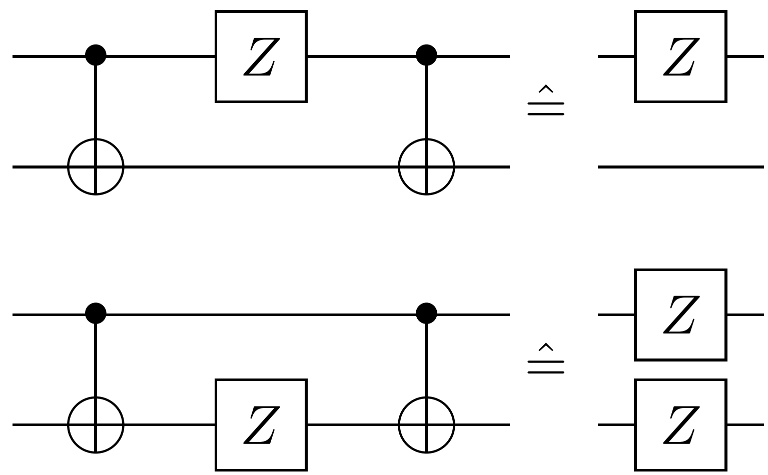


Action of CNOT

Modifying stabilizers



add X from source to target



add Z from target to source

Example: 5 Qubit Code $[\![5,1,3]\!]$

Generators of stabilizer

$$\left[\begin{array}{c} XXZIZ \\ ZXXZI \\ IZXXZ \\ ZIZXX \end{array}\right] \mathrel{\hat{=}} \left(\begin{array}{ccccc|ccccc} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 \end{array}\right)$$

Step I: X-Only Generator

Local operations $\hat{=}$ operations on corresponding X and Z columns

$$\left(\begin{array}{ccccc|ccccc} 1 & 1 & {\color{blue}0} & 0 & {\color{blue}0} & 0 & 0 & {\color{red}1} & 0 & {\color{red}1} \\ 0 & 1 & {\color{blue}1} & 0 & {\color{blue}0} & 1 & 0 & {\color{red}0} & 1 & {\color{red}0} \\ 0 & 0 & {\color{blue}1} & 1 & {\color{blue}0} & 0 & 1 & {\color{red}0} & 0 & {\color{red}1} \\ 0 & 0 & {\color{blue}0} & 1 & {\color{blue}1} & 1 & 0 & {\color{red}1} & 0 & {\color{red}0} \end{array}\right)$$

$$T_1 := I \otimes I \otimes I \otimes I \otimes I$$

Step I: X-Only Generator

Local operations $\hat{=}$ operations on corresponding X and Z columns

$$\left(\begin{array}{ccccc|ccccc} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \end{array}\right)$$

$$T_1 := I \otimes I \otimes H \otimes I \otimes I$$

Step I: X-Only Generator

Local operations $\hat{=}$ operations on corresponding X and Z columns

$$\left(\begin{array}{ccccc|ccccc}
1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\
0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 \\
0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1
\end{array}\right)$$

$$T_1 := I \otimes I \otimes H \otimes I \otimes H$$

Step II: X-Generator of Weight One

$\mathrm{CNOT} \mathrel{\hat{=}}$ operations on pairs of corresponding X and Z columns

$$\left(\begin{array}{ccccc|ccccc} 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{array}\right)$$

$$T_1 := I \otimes I \otimes H \otimes I \otimes H$$

$$T_2 := I$$

Step II: X-Generator of Weight One

$\mathrm{CNOT} \mathrel{\hat{=}}$ operations on pairs of corresponding X and Z columns

$$\left(\begin{array}{ccccc|ccccc} \color{red}{1} & 1 & 1 & 0 & 1 & 0 & \color{blue}{0} & 0 & 0 & 0 \\ \color{red}{0} & 1 & 0 & 0 & 0 & 1 & \color{blue}{0} & 1 & 1 & 0 \\ \color{red}{0} & 0 & 0 & 1 & 1 & 0 & \color{blue}{1} & 1 & 0 & 0 \\ \color{red}{0} & 0 & 1 & 1 & 0 & 1 & \color{blue}{0} & 0 & 0 & 1 \end{array}\right)$$

$$T_1 := I \otimes I \otimes H \otimes I \otimes H$$

$$T_2 := I$$

Step II: X-Generator of Weight One

$\mathrm{CNOT} \mathrel{\hat{=}}$ operations on pairs of corresponding X and Z columns

$$\left(\begin{array}{ccccc|ccccc} 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{array}\right)$$

$$T_1 := I \otimes I \otimes H \otimes I \otimes H$$

$$T_2 := \mathrm{CNOT}^{(1,2)}$$

Step II: X-Generator of Weight One

$\mathrm{CNOT} \mathrel{\hat{=}}$ operations on pairs of corresponding X and Z columns

$$\left(\begin{array}{ccccc|ccccc} 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{array}\right)$$

$$T_1 := I \otimes I \otimes H \otimes I \otimes H$$

$$T_2 := \mathrm{CNOT}^{(1,2)}$$

Step II: X-Generator of Weight One

$\mathrm{CNOT} \hat{=}$ operations on pairs of corresponding X and Z columns

$$\left(\begin{array}{ccccc|ccccc} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{array}\right)$$

$$T_1 := I \otimes I \otimes H \otimes I \otimes H$$

$$T_2 := \mathrm{CNOT}^{(1,2)}\mathrm{CNOT}^{(1,3)}$$

Step II: X-Generator of Weight One

$\mathrm{CNOT} \hat{=}$ operations on pairs of corresponding X and Z columns

$$\left(\begin{array}{ccccc|ccccc} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{array}\right)$$

$$T_1 := \ I \otimes I \otimes H \otimes I \otimes H$$

$$T_2 := \mathrm{CNOT}^{(1,2)}\mathrm{CNOT}^{(1,3)}$$

Step II: X-Generator of Weight One

$\mathrm{CNOT} \hat{=}$ operations on pairs of corresponding X and Z columns

$$\left(\begin{array}{ccccc|ccccc} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \end{array}\right)$$

$$T_1 := \ I \otimes I \otimes H \otimes I \otimes H$$

$$T_2 := \mathrm{CNOT}^{(1,2)}\mathrm{CNOT}^{(1,3)}\mathrm{CNOT}^{(1,5)}$$

Step III: Row Operations

multiplying generators $\hat{=}$ adding/permuting rows

$$\left(\begin{array}{ccccc|ccccc} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \end{array}\right)$$

$$T_1 := I \otimes I \otimes H \otimes I \otimes H$$

$$T_2 := \mathrm{CNOT}^{(1,2)}\mathrm{CNOT}^{(1,3)}\mathrm{CNOT}^{(1,5)}$$

Step III: Row Operations

multiplying generators $\hat{=}$ adding/permuting rows

$$\left(\begin{array}{ccccc|ccccc} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 \end{array}\right)$$

$$T_1 := I \otimes I \otimes H \otimes I \otimes H$$

$$T_2 := \mathrm{CNOT}^{(1,2)}\mathrm{CNOT}^{(1,3)}\mathrm{CNOT}^{(1,5)}$$

Step IV: Next Row/Generator

$$\left(\begin{array}{ccccc|ccccc}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\
0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0
\end{array}\right)$$

$$T_3 := \; I \otimes I \otimes I \otimes I \otimes I$$

Step IV: Next Row/Generator

$$\left(\begin{array}{ccccc|ccccc} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \end{array}\right)$$

$$T_3 := I \otimes I \otimes H \otimes H \otimes I$$

Step IV: Next Row/Generator

$$\left(\begin{array}{ccccc|ccccc} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \end{array}\right)$$

$$T_3 := I \otimes I \otimes H \otimes H \otimes I$$

$$T_4 := \mathrm{CNOT}^{(2,3)}\mathrm{CNOT}^{(2,4)}$$

Step IV: Next Row/Generator

$$
\left(\begin{array}{ccccc|ccccc}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1
\end{array}\right)
$$

$$T_3 := I \otimes I \otimes H \otimes H \otimes I$$

$$T_4 := \mathrm{CNOT}^{(2,3)} \mathrm{CNOT}^{(2,4)}$$

Pre-Final Result: Only X-Generators

$$\left(\begin{array}{ccccc|ccccc}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0
\end{array}\right)$$

Final Result: Only Z-Generators

$$\left(\begin{array}{ccccc|ccccc} 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{array}\right)$$

$$T_{\text{final}} := H \otimes H \otimes H \otimes H \otimes I$$

Final Result: Only Z-Generators

$$\left(\begin{array}{ccccc|ccccc} 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{array}\right)$$

$$T_{\text{final}} := H \otimes H \otimes H \otimes H \otimes I$$

combining all transformations: quantum circuit that maps encoded state $|\psi_L\rangle$ to the un-encoded state $|0\rangle|\psi\rangle$

Final Result: Only Z-Generators

$$\left(\begin{array}{ccccc|ccccc} 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{array}\right)$$

$$T_{\text{final}} := H \otimes H \otimes H \otimes H \otimes I$$

combining all transformations: quantum circuit that maps encoded state $|\psi_L\rangle$ to the un-encoded state $|0\rangle|\psi\rangle$

“Homework”: derive the complete circuit

Decoding Circuit for Ternary QECC $\mathcal{C} = [\![9,5,3]\!]_3$