Center for
Quantum Networks
NSF Engineering Research Center

Classical and Quantum Error Correction

Instructor: Narayanan Rengaswamy
– University of Arizona

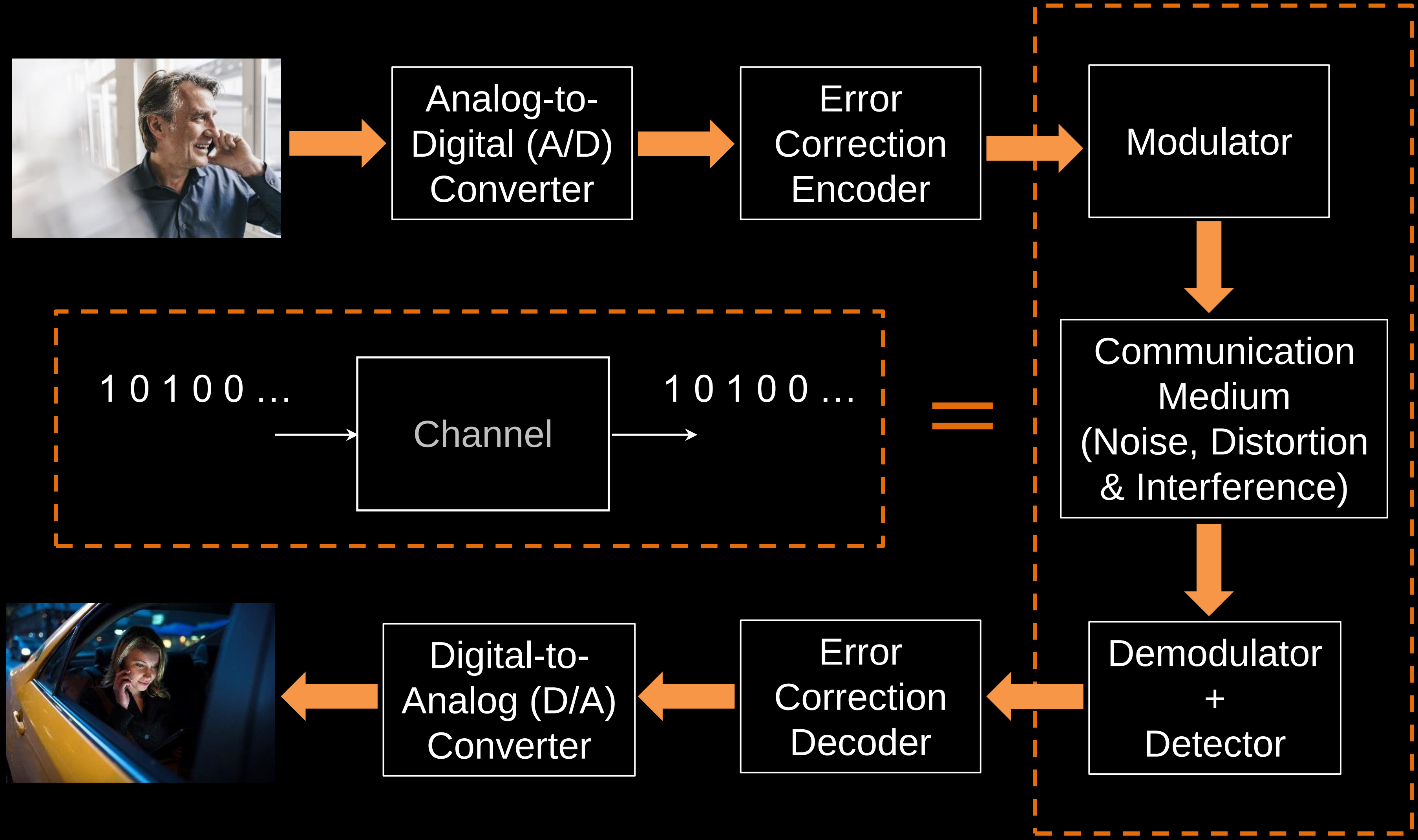
Instructor: Bane Vasić
— University of Arizona

CQN Winter School on Quantum Networks

Funded by National Science Foundation Grant #1941583

Prelude

Digital communications and storage



Analog-to-Digital (A/D) Converter

Error Correction Encoder

Modulator

Communication Medium (Noise, Distortion & Interference)

Demodulator + Detector

Error Correction Decoder

Digital-to-Analog (D/A) Converter

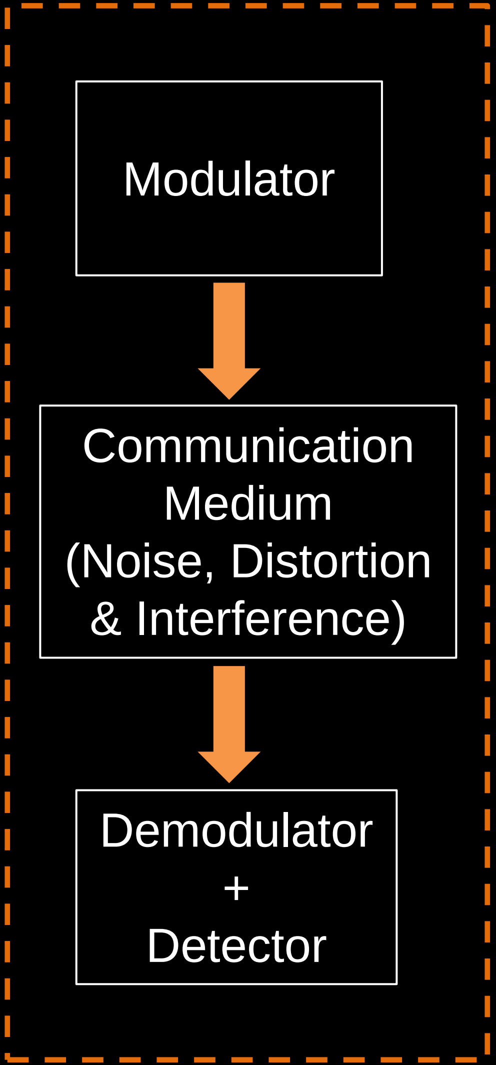
1 0 1 0 0 …

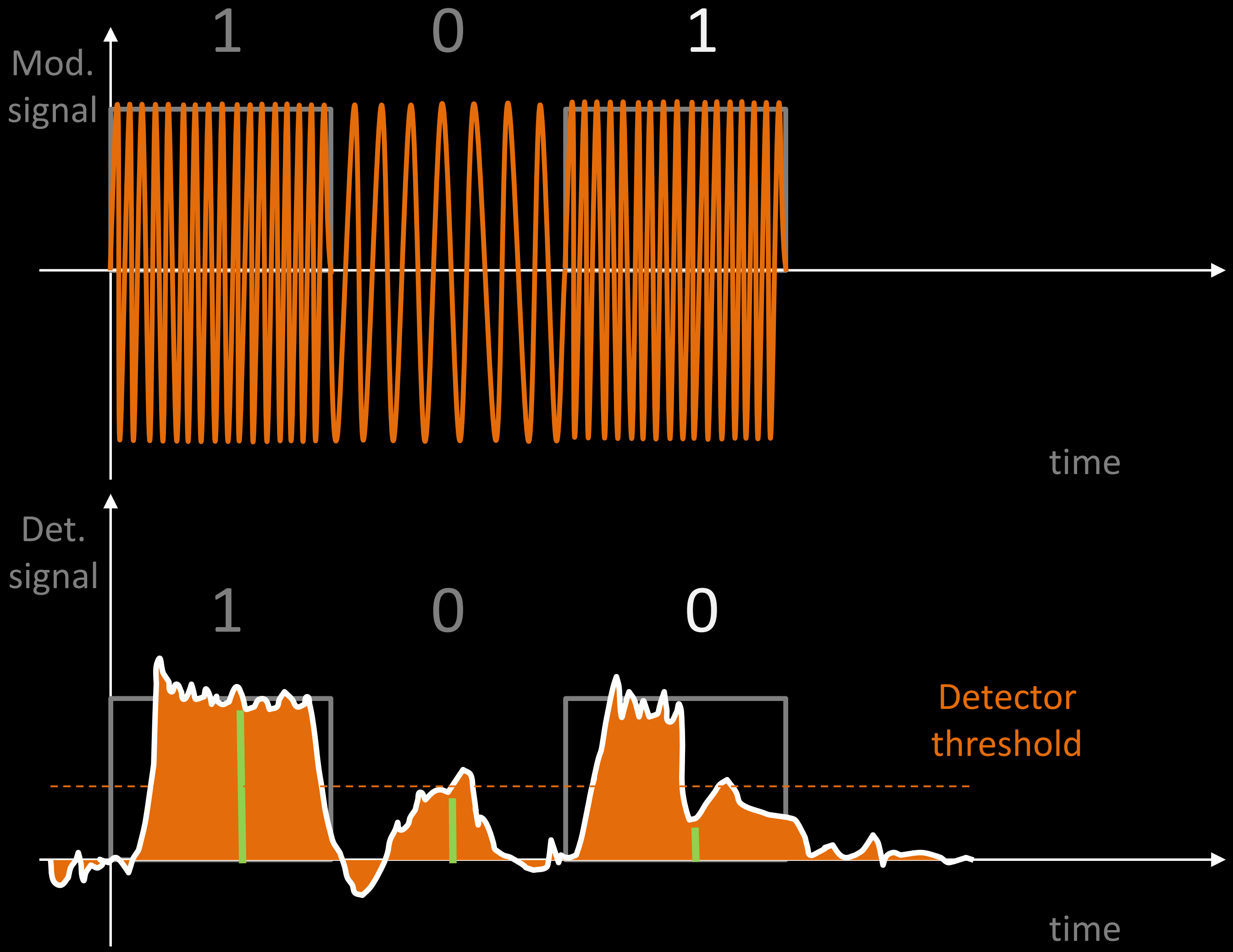
Channel

1 0 1 0 0 …

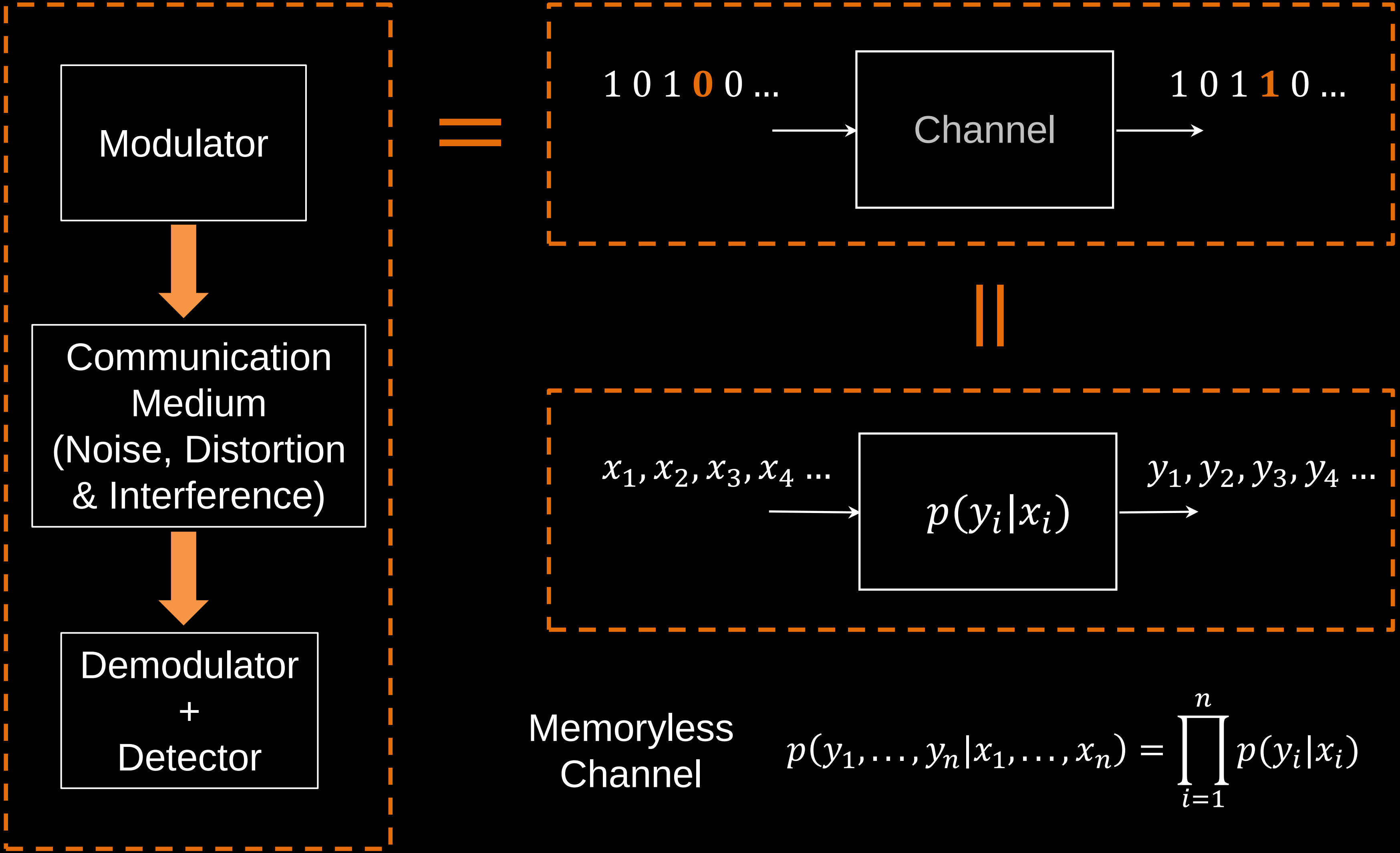
=

Inside the box: Channel + Detector errors





Noisy memoryless channels

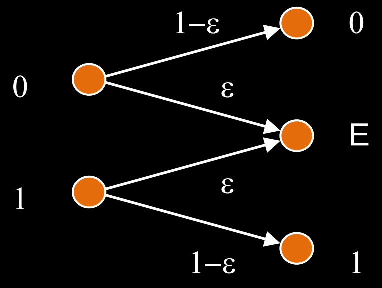


Memoryless Channel

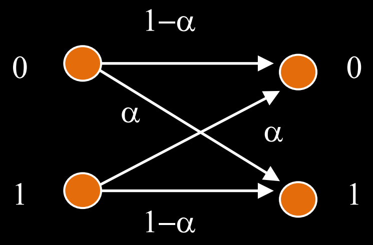
$$p(y_1,\ldots,y_n|x_1,\ldots,x_n)=\prod_{i=1}^{n}p(y_i|x_i)$$

Simple memoryless channels

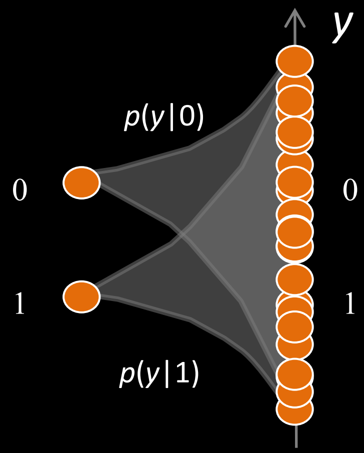
- Binary erasure channel (BEC)



- Binary symmetric channel (BSC)

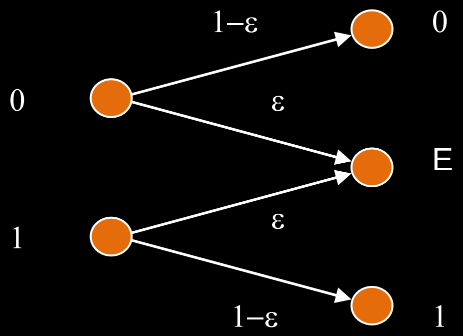


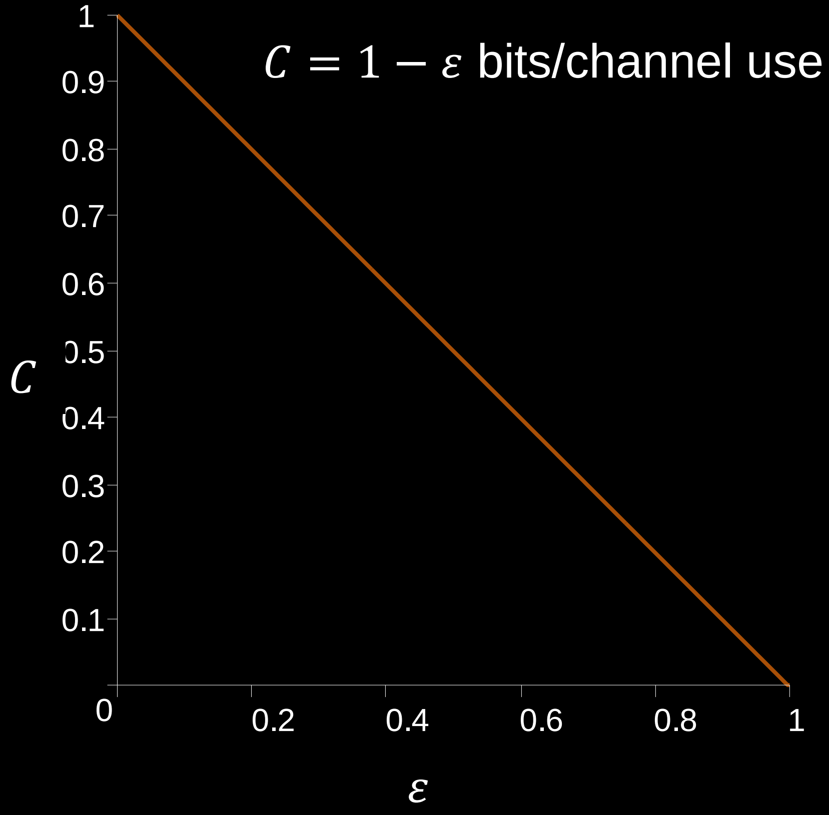
- Binary input additive white Gaussian noise (AWGN) channel, σ^2



Channel capacity – Binary Erasure Channel

Channel Capacity





We lose a fraction ε of the bits. How do we recover that data?

Poll Question 1

Let $\boldsymbol{x} = [x_1, \ldots, x_n]$ be a codeword transmitted over a memoryless channel and let $\boldsymbol{y} = [y_1, \ldots, y_n]$ be the corresponding channel output. Then the conditional probability density $\boldsymbol{p}(\boldsymbol{y}|\boldsymbol{x})$ can be written as

A. $p(\boldsymbol{y}|\boldsymbol{x}) = \sum_{i=1}^{n} p(y_i|x_i)$

B. $p(\boldsymbol{y}|\boldsymbol{x}) = p(y_1|x_1)\, p(y_2|x_2)\, p(y_3|x_3) \cdots p(y_{n-1}|x_{n-1})\, p(y_n|x_n)$

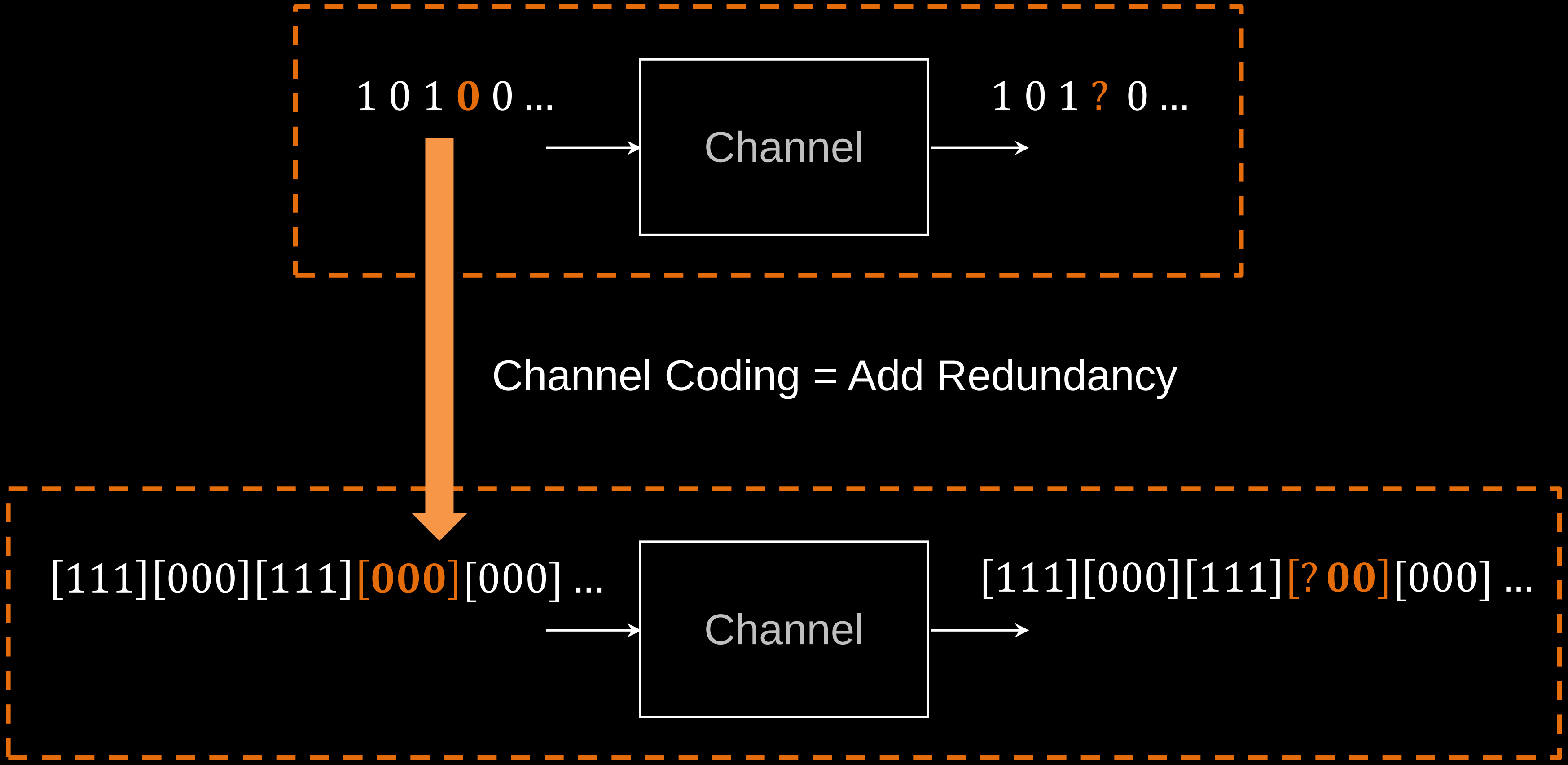
C. $p(\boldsymbol{y}|\boldsymbol{x}) = p(y_1|x_2) + p(y_2|x_3) + p(y_3|x_4) + \cdots + p(y_{n-1}|x_n)$

D. $p(\boldsymbol{y}|\boldsymbol{x}) = p(y_1|x_2) \cup p(y_2|x_3) \cup p(y_3|x_4) \cup \cdots \cup p(y_{n-1}|x_n)$

E. $p(\boldsymbol{y}|\boldsymbol{x}) = p(\{y_1 + y_2 + , \ldots + y_n\} \cup \{x_1 + x_2 + \cdots + x_n\})$

F. None of the above

Channel coding



1 0 1 **0** 0 ...

Channel

1 0 1 ? 0 ...

Channel Coding = Add Redundancy

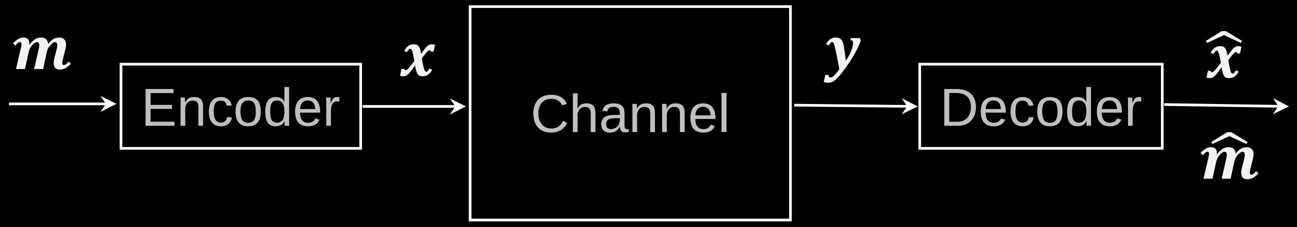
[111][000][111]**[000]**[000] ...

Channel

[111][000][111][? **00**][000] ...

$C = 1 - \varepsilon$: But the n-bit repetition code sends just 1 bit / n channel uses!

Error Correction Coding (ECC)



- Message: $\boldsymbol{m} = [m_1, m_2, \dots, m_k]$
- Codeword: $\boldsymbol{x} = [x_1, x_2, \dots, x_n]$
- Code rate: $R = \frac{k}{n} \leq C$ (Capacity) ≤ 1
- Received word: $\boldsymbol{y} = [y_1, y_2, \dots, y_n]$

- The decoder tries to find $\hat{\boldsymbol{x}}$ (or $\hat{\boldsymbol{m}}$) from $\boldsymbol{y}$ so that the probability of bit/codeword error is minimal.
- In other words, decoder tries to find a codeword that is “closest” to $\boldsymbol{y}$.

$[n, k]$ binary linear codes

Generator matrix $(k \times n)$: $G = \begin{bmatrix} \boldsymbol{g_1} \\ \boldsymbol{g_2} \\ \vdots \\ \boldsymbol{g_k} \end{bmatrix} \in \{0,1\}^{k \times n}$ (rank k binary matrix)

Encoding: $\boldsymbol{x} = \boldsymbol{m}G = m_1\boldsymbol{g_1} + m_2\boldsymbol{g_2} + \cdots + m_k\boldsymbol{g_k} \in \{0,1\}^n$ (XOR)

---

n-bit Repetition Code: $G = [\ \boldsymbol{g}\ ] = [1 \quad 1 \quad 1 \quad \cdots \quad 1]$

$$\boldsymbol{m} = [\ m\ ] \xrightarrow{Encode} \boldsymbol{x} = \boldsymbol{m}G = [m \quad m \quad m \quad \cdots \quad m]$$

$[n = 5, k = 2]$ Code: (contains $2^k = 4$ codewords to encode $2^k = 4$ messages)

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

$$\begin{aligned}
\boldsymbol{m} &= [0 \quad 0] \xrightarrow{Encode} \boldsymbol{x} = [0 \quad 0 \quad 0 \quad 0 \quad 0] \\
\boldsymbol{m} &= [0 \quad 1] \xrightarrow{Encode} \boldsymbol{x} = [0 \quad 1 \quad 1 \quad 1 \quad 0] \\
\boldsymbol{m} &= [1 \quad 0] \xrightarrow{Encode} \boldsymbol{x} = [1 \quad 0 \quad 1 \quad 0 \quad 1] \\
\boldsymbol{m} &= [1 \quad 1] \xrightarrow{Encode} \boldsymbol{x} = [1 \quad 1 \quad 0 \quad 1 \quad 1]
\end{aligned}$$

Parity-check matrix

[$n = 5, k = 2$] Code: (contains $2^k = 4$ codewords to encode $2^k = 4$ messages)

$$G = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 \end{bmatrix} = \begin{bmatrix} \boldsymbol{g}_1 \\ \boldsymbol{g}_2 \\ \vdots \\ \boldsymbol{g}_k \end{bmatrix}$$

$$\begin{aligned} \boldsymbol{m} &= [0 \quad 0] \xrightarrow{Encode} \boldsymbol{x} = [0 \quad 0 \quad 0 \quad 0 \quad 0] \\ \boldsymbol{m} &= [0 \quad 1] \xrightarrow{Encode} \boldsymbol{x} = [0 \quad 1 \quad 1 \quad 1 \quad 0] \\ \boldsymbol{m} &= [1 \quad 0] \xrightarrow{Encode} \boldsymbol{x} = [1 \quad 0 \quad 1 \quad 0 \quad 1] \\ \boldsymbol{m} &= [1 \quad 1] \xrightarrow{Encode} \boldsymbol{x} = [1 \quad 1 \quad 0 \quad 1 \quad 1] \end{aligned}$$

Encoding:
$$\begin{aligned} \boldsymbol{x} = [m_1 \quad m_2]G &= [m_1 \quad m_2 \quad m_1 + m_2 \quad m_2 \quad m_1] \\ &= [x_1 \quad x_2 \quad x_3 \quad x_4 \quad x_5] \end{aligned}$$

Parity-checks:
$$\begin{aligned} s_1 &= x_1 + x_2 + x_3 = m_1 + m_2 + (m_1 + m_2) = 0 \\ s_2 &= x_2 + x_4 = m_2 + m_2 = 0 \\ s_3 &= x_1 + x_5 = m_1 + m_1 = 0 \end{aligned}$$

$$\boxed{(HG^T)m^T = 0 \Rightarrow HG^T = 0}$$

Syndrome

Parity-check matrix $(n - k) \times n$:
$$H = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \boldsymbol{h}_1 \\ \boldsymbol{h}_2 \\ \vdots \\ \boldsymbol{h}_{n-k} \end{bmatrix}; \boldsymbol{s} = \begin{bmatrix} s_1 \\ s_2 \\ s_3 \end{bmatrix} = H\boldsymbol{x}^T = H \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Poll Question 2

For an $[n, k]$ linear block code, what are the dimensions of the generator and parity-check matrices?

$A.$ G: $kn \times n$ and H: $(n-k)n \times n$

$B.$ G: $k \times n$ and H: $(n-k) \times n$

$C.$ G: $k \times k$ and H: $(n-k) \times (n-k)$

$D.$ G: $n \times k$ and H: $n \times (n-k)$

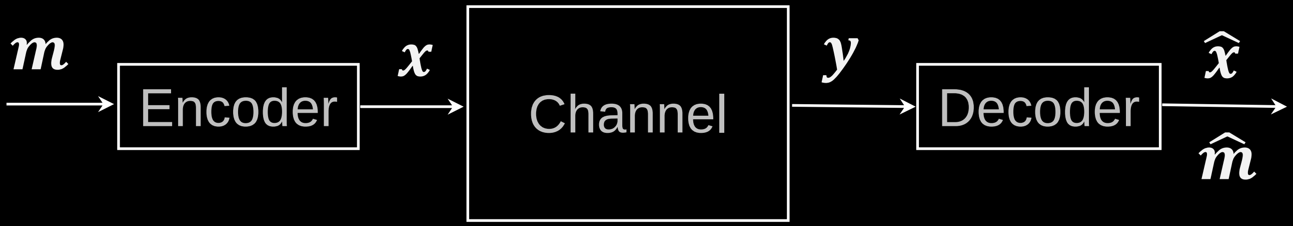
Correcting errors

Codebook

$$\boldsymbol{x} = [0 \quad 0 \quad 0 \quad 0 \quad 0]$$
$$\boldsymbol{x} = [0 \quad 1 \quad 1 \quad 1 \quad 0]$$
$$\boldsymbol{x} = [1 \quad 0 \quad 1 \quad 0 \quad 1]$$
$$\boldsymbol{x} = [1 \quad 1 \quad 0 \quad 1 \quad 1]$$

Syndrome

$$H = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \end{bmatrix} ; \boldsymbol{s} = H\boldsymbol{x}^{\boldsymbol{T}} = \boldsymbol{0}$$



Channel = BEC: $\boldsymbol{y} = [1 \quad E \quad 0 \quad 1 \quad 1] \overset{Decode}{\Longrightarrow} \widehat{\boldsymbol{x}} = [1 \quad 1 \quad 0 \quad 1 \quad 1]$

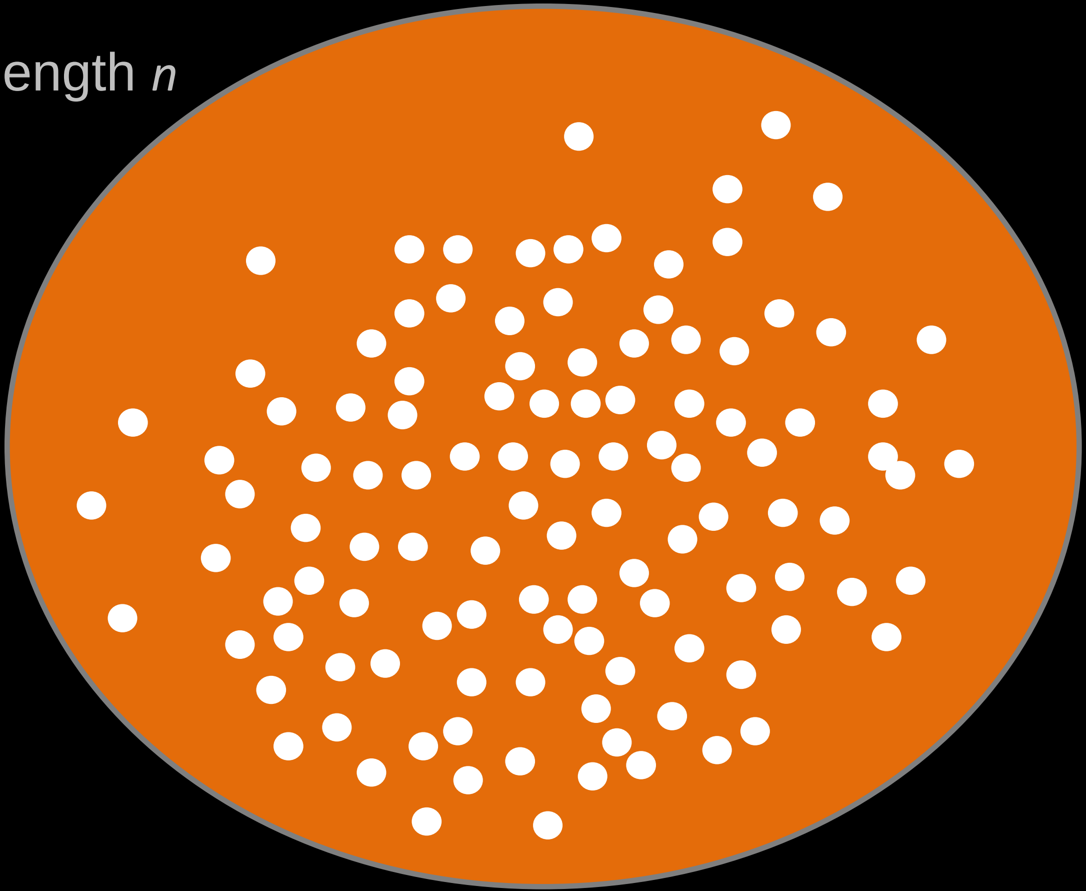
$\boldsymbol{y} = [1 \quad E \quad E \quad 1 \quad 1] \overset{Decode}{\Longrightarrow} \widehat{\boldsymbol{x}} = [1 \quad 1 \quad 0 \quad 1 \quad 1]$

Channel = BSC: $\boldsymbol{e} = [0 \quad 1 \quad 0 \quad 0 \quad 0] ; \boldsymbol{s} = H\boldsymbol{y}^{\boldsymbol{T}} = H(\boldsymbol{x} + \boldsymbol{e})^{T} = H\boldsymbol{e}^{\boldsymbol{T}} = [1 \quad 1 \quad 0]^{T}$

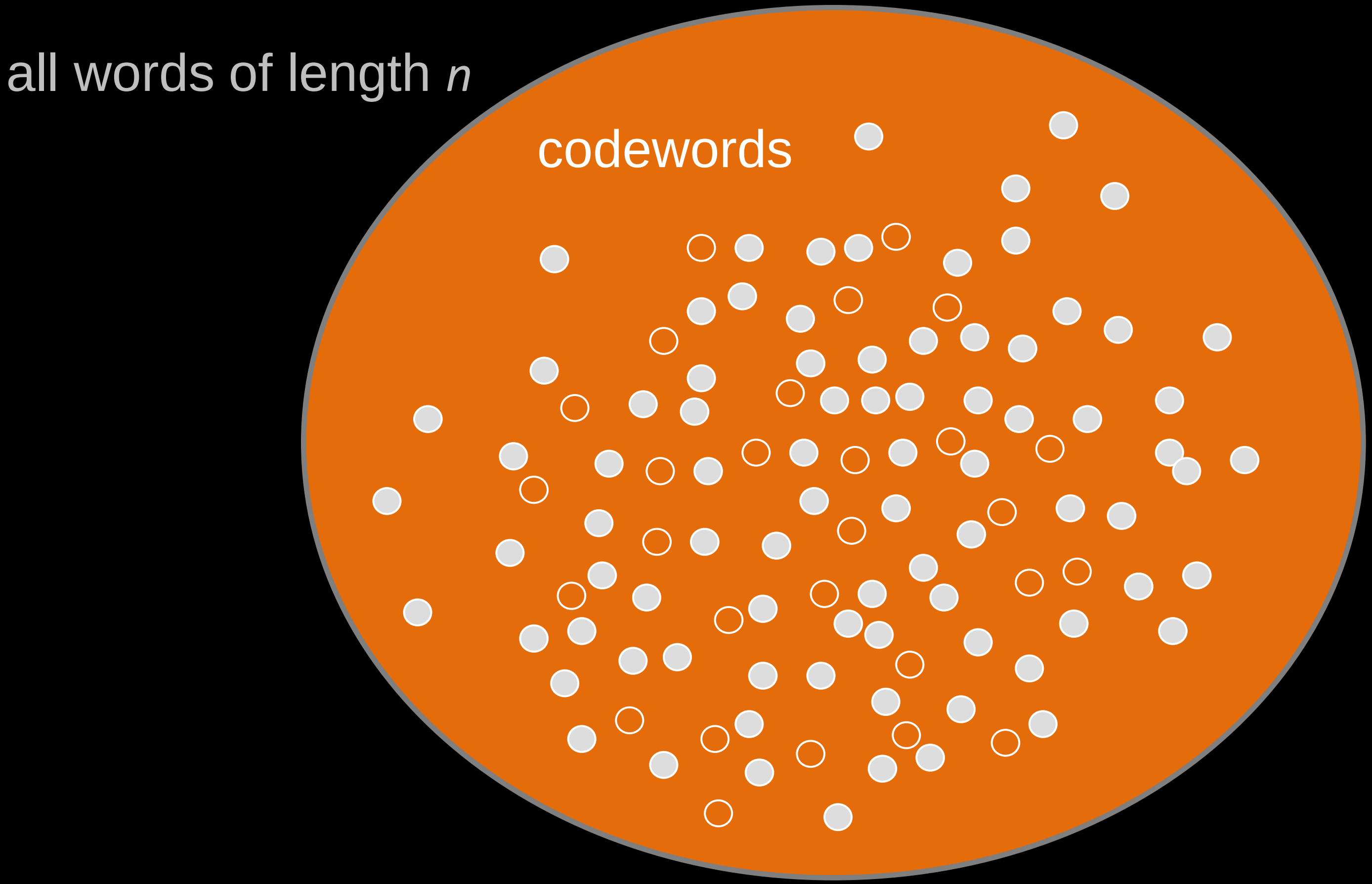
$\boldsymbol{y} = [1 \quad 0 \quad 0 \quad 1 \quad 1] \overset{Decode}{\Longrightarrow} \widehat{\boldsymbol{x}} = \boldsymbol{y} + \boldsymbol{e} = [1 \quad 1 \quad 0 \quad 1 \quad 1]$

Protecting information by coding

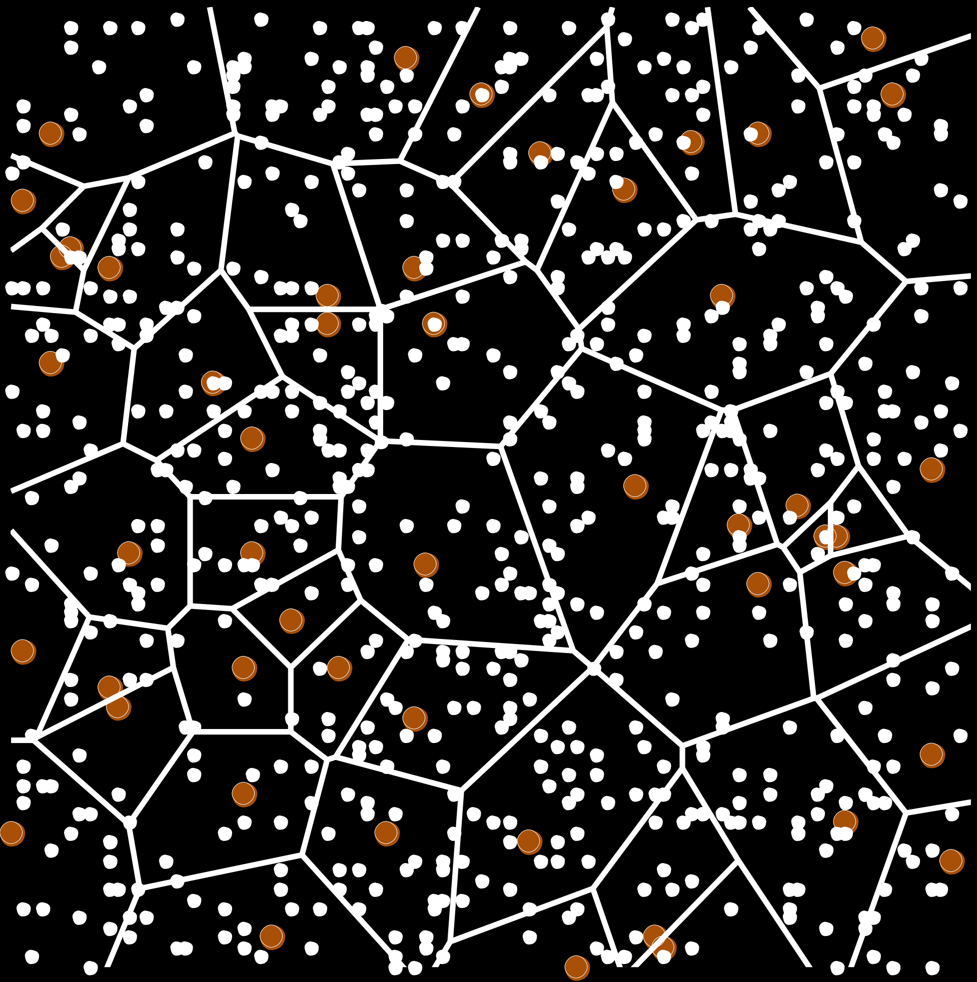
all words of length *n*



Protecting information by coding



Maximum likelihood decoding



Poll Question 3

Consider the code with the following parity-check matrix

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

and assume that the channel output is $\boldsymbol{y} = [1\ 1\ 0\ 0\ 0]$. The syndrome is then:

A. $\boldsymbol{s} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$

B. $\boldsymbol{s} = \begin{bmatrix} 2 \\ 1 \\ 1 \end{bmatrix}$

C. $\boldsymbol{s} = 3$

D. $\boldsymbol{s} = 6$

E. $\boldsymbol{s} = 2^3$

F. None of the above

G. I’m not sure

Poll Question 4

Consider the code with the following parity-check matrix

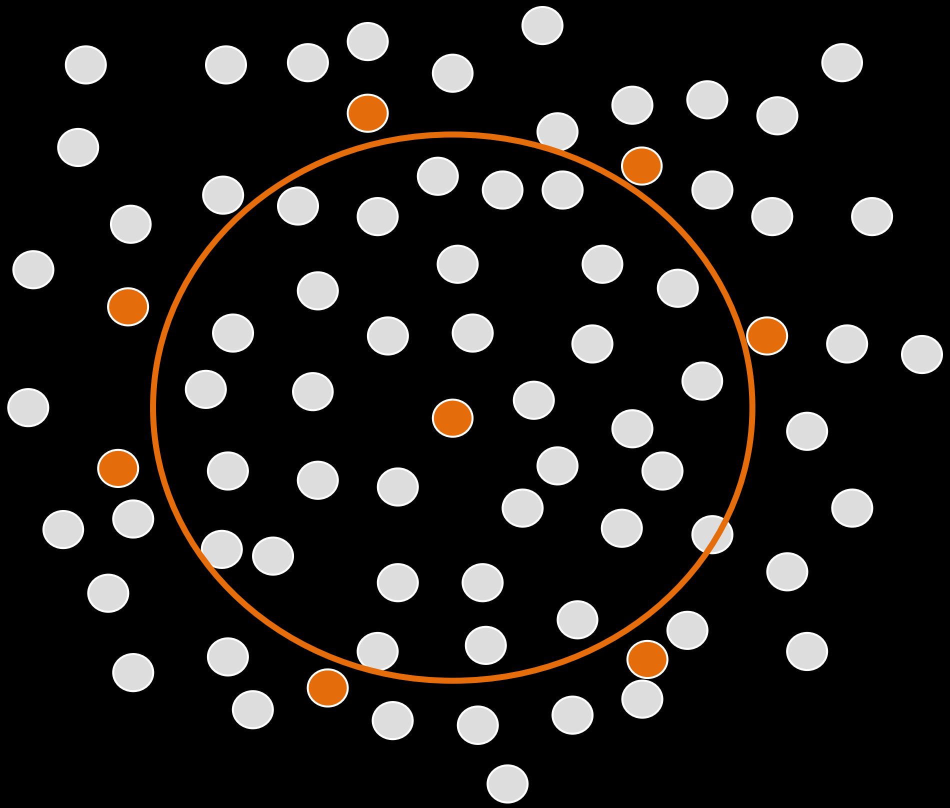
$$H = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

and assume that the channel output is $\boldsymbol{y} = [0\ 0\ 0\ 0\ E]$, where E is the erasure symbol. The erasure decoder will produce the following vector as the output:

A. $\boldsymbol{x} = [0\ 0\ 0\ 0\ 0]$
B. $\boldsymbol{x} = [0\ 0\ 0\ 0\ 1]$
C. $\boldsymbol{e} = [0\ 0\ 0\ 0\ 1]$
D. $\boldsymbol{m} = [0\ 0\ 1]$
E. $\boldsymbol{m} = [0\ 1]$
F. None of the above

G. I'm not sure

Minimum distance



Hamming distance = the number of positions in which two binary vectors differ
Minimum distance = the Hamming distance between two closest codewords

$[n = 5, k = 2, d = 3]$ linear block code

Codebook

$$\begin{aligned} \boldsymbol{x_1} &= [0 \quad 0 \quad 0 \quad 0 \quad 0] \\ \boldsymbol{x_2} &= [0 \quad 1 \quad 1 \quad 1 \quad 0] \\ \boldsymbol{x_3} &= [1 \quad 0 \quad 1 \quad 0 \quad 1] \\ \boldsymbol{x_4} &= [1 \quad 1 \quad 0 \quad 1 \quad 1] \end{aligned}$$

Syndrome

$$H = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \end{bmatrix}; \boldsymbol{s} = H\boldsymbol{x}^{\boldsymbol{T}} = \boldsymbol{0}$$

Hamming distance = the number of positions in which two binary vectors differ

Minimum distance d = the Hamming distance between two closest codewords

= the Hamming distance between $\boldsymbol{x_1}$ and $\boldsymbol{x_2}$

= the Hamming distance between $\boldsymbol{x_1}$ and $\boldsymbol{x_3}$

= the Hamming distance between $\boldsymbol{x_2}$ and $\boldsymbol{x_4}$

= the Hamming distance between $\boldsymbol{x_3}$ and $\boldsymbol{x_4}$

= 3

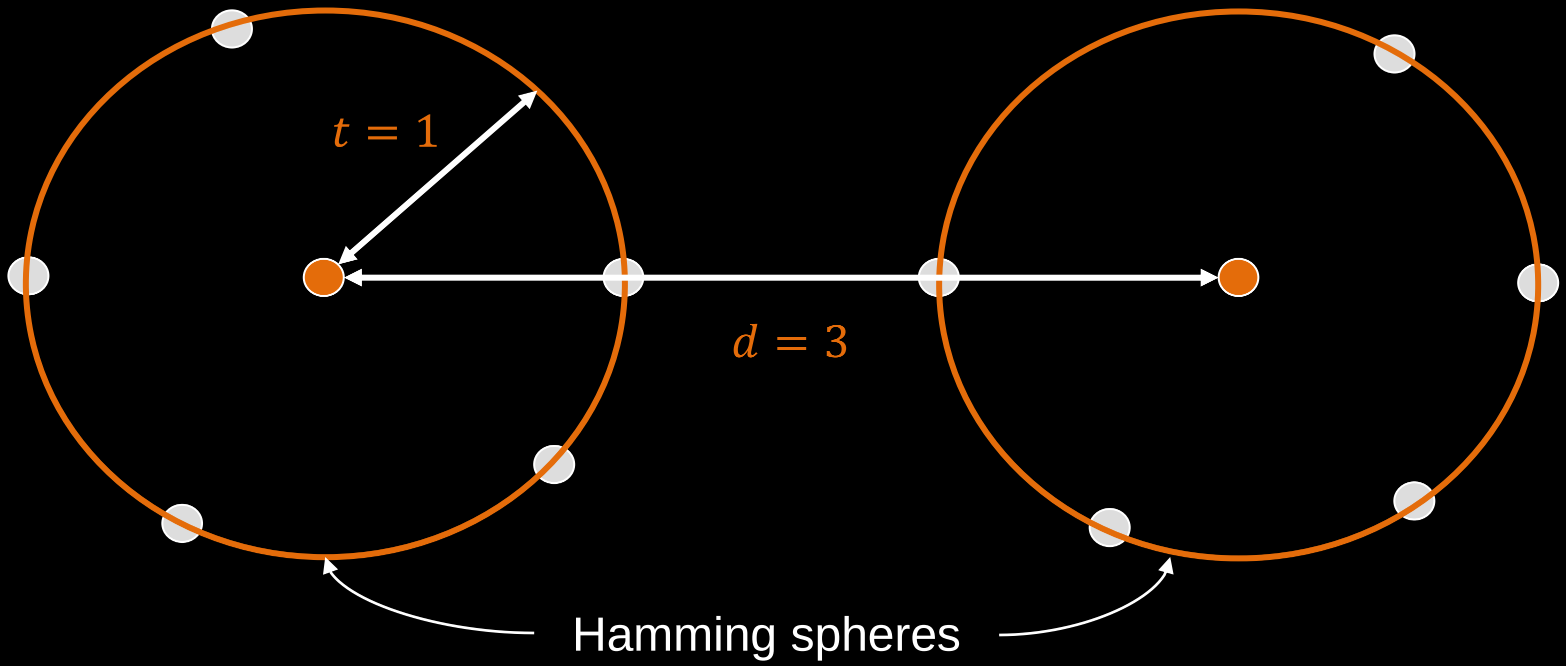
The code encodes $k = 2$ message bits into $n = 5$ code bits with distance $d = 3$

Hamming spheres of radius $t = \frac{d-1}{2} = 1$ around codewords don’t intersect

➡ Code corrects $t = 1$ error or $d - 1 = 2$ erasures; detects $d - 1 = 2$ errors

Minimum distance

$\binom{n}{t} = 5$ vectors at Hamming distance $t = 1$ from any codeword



Hamming distance = the number of positions in which two binary vectors differ
Minimum distance = the Hamming distance between two closest codewords

Dual code $C^{\perp}$

Generator and Parity-check

$$G = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 \end{bmatrix} \quad H = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \end{bmatrix}; \boldsymbol{s} = H\boldsymbol{x}^{\boldsymbol{T}} = \mathbf{0}$$

Syndrome

Duality (Orthogonality): $GH^T = 0$

Dual Code

Row space of H: $x_i v_j^T = 0$

Code C

$$\begin{aligned} \boldsymbol{x_1} &= [0 \quad 0 \quad 0 \quad 0 \quad 0] \\ \boldsymbol{x_2} &= [1 \quad 0 \quad 1 \quad 0 \quad 1] \\ \boldsymbol{x_3} &= [0 \quad 1 \quad 1 \quad 1 \quad 0] \\ \boldsymbol{x_4} &= [1 \quad 1 \quad 0 \quad 1 \quad 1] \end{aligned}$$

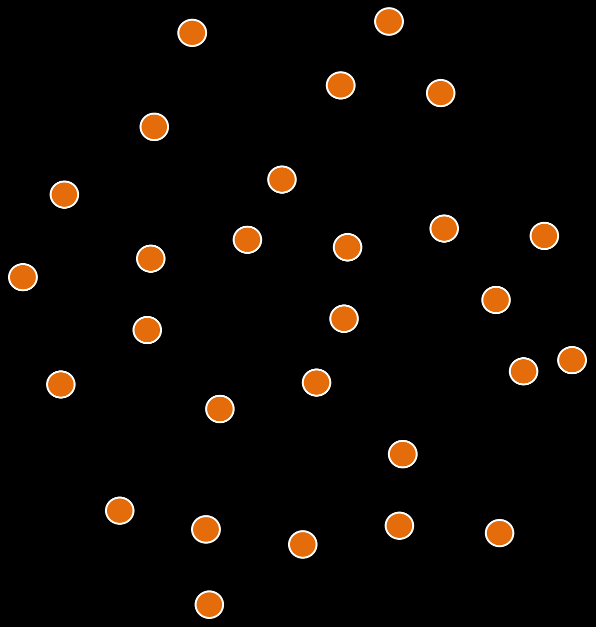
Code $C^{\perp}$

$$\begin{aligned} \boldsymbol{v_1} &= [0 \quad 0 \quad 0 \quad 0 \quad 0] \\ \boldsymbol{v_2} &= [1 \quad 1 \quad 1 \quad 0 \quad 0] \\ \boldsymbol{v_3} &= [0 \quad 1 \quad 0 \quad 1 \quad 0] \\ \boldsymbol{v_4} &= [1 \quad 0 \quad 0 \quad 0 \quad 1] \\ \boldsymbol{v_5} &= [1 \quad 0 \quad 1 \quad 1 \quad 0] \\ \boldsymbol{v_6} &= [0 \quad 1 \quad 1 \quad 0 \quad 1] \\ \boldsymbol{v_7} &= [1 \quad 1 \quad 0 \quad 1 \quad 1] \\ \boldsymbol{v_8} &= [0 \quad 0 \quad 1 \quad 1 \quad 1] \end{aligned}$$

Code C : $[n = 5, k = 2, d = 3]$

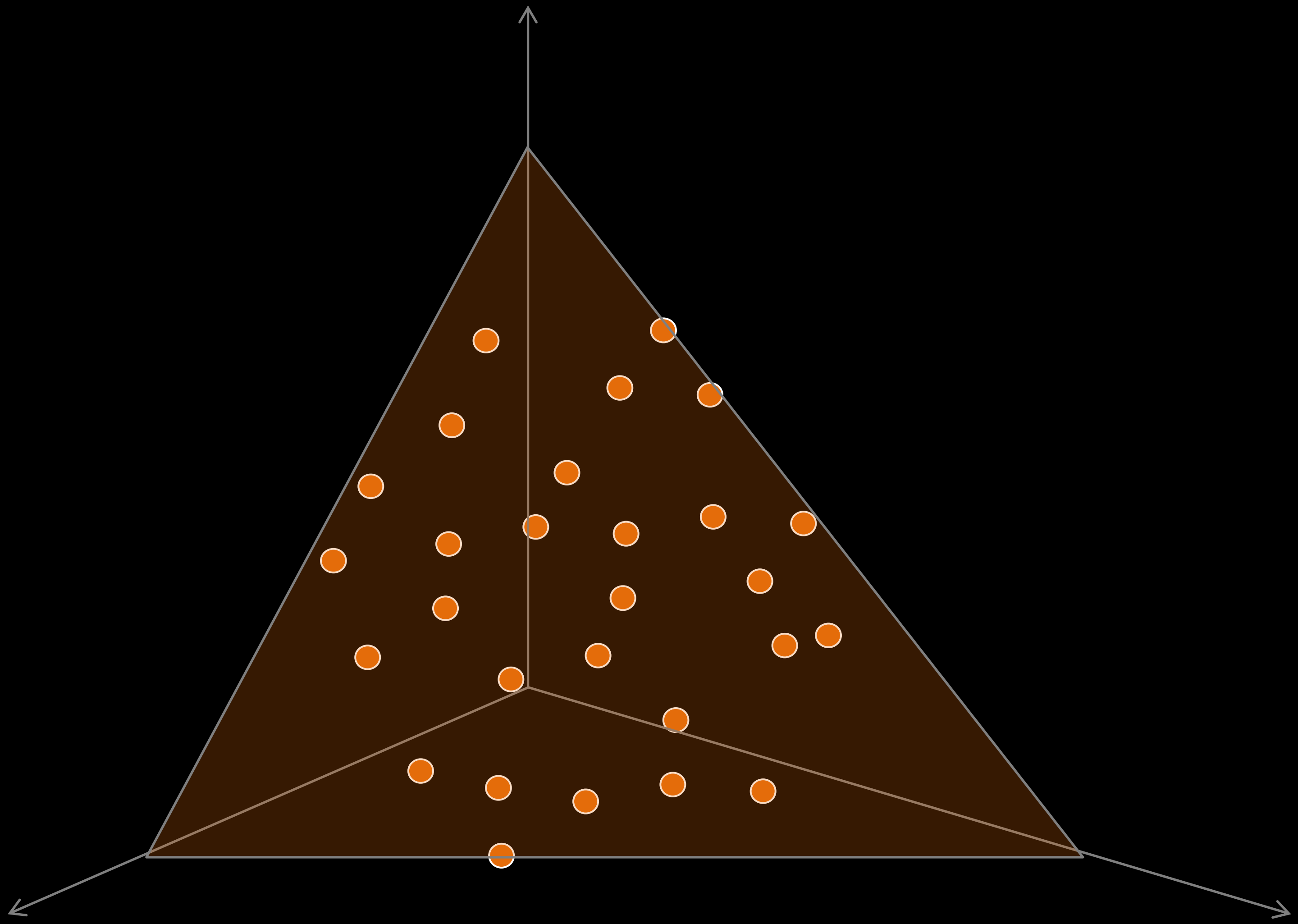
Code $C^{\perp}$: $[n = 5, k^{\perp} = n - k = 3, d^{\perp} = 2]$

Protecting information by coding



Center for Quantum Networks *NSF-ERC*

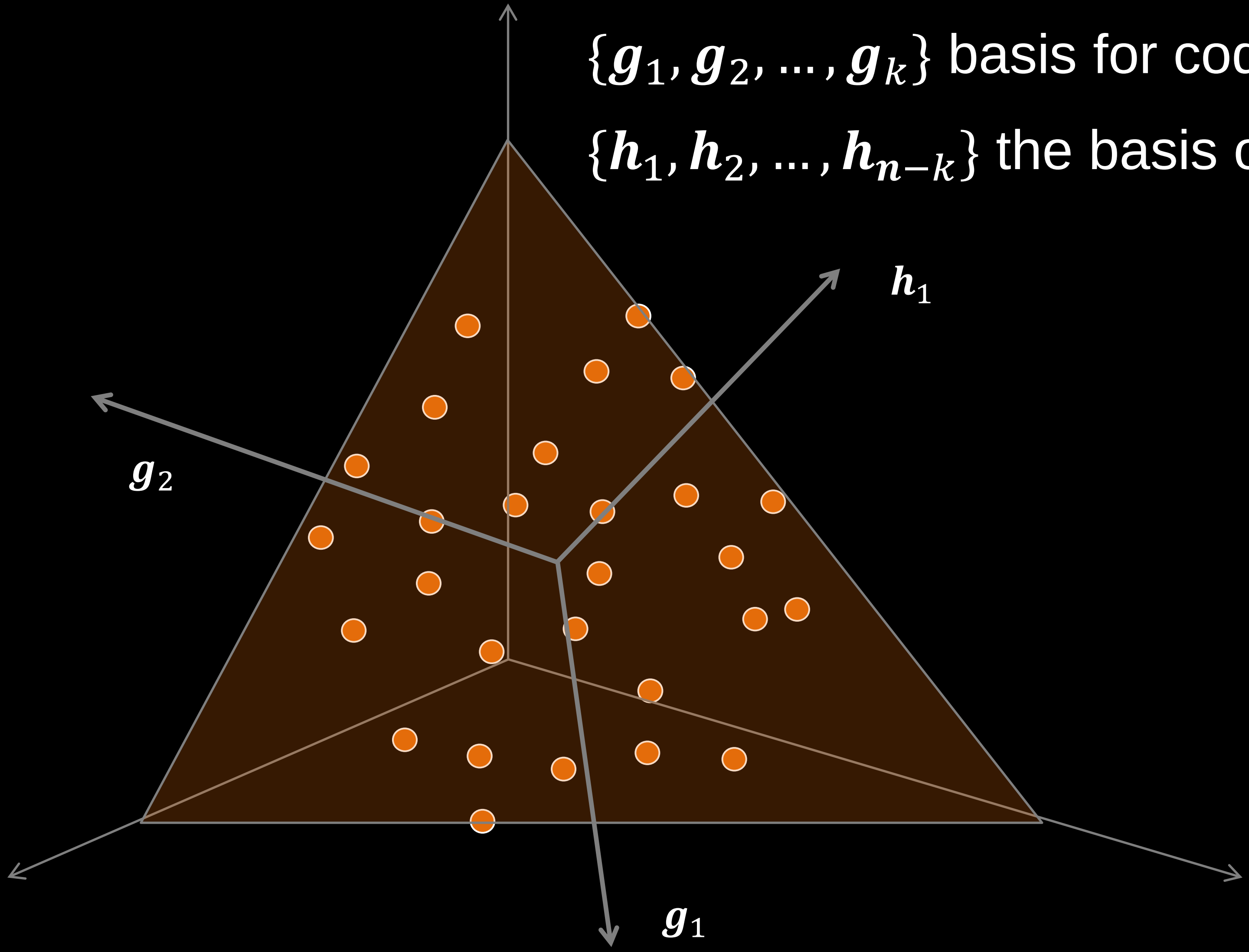
Linear block codes



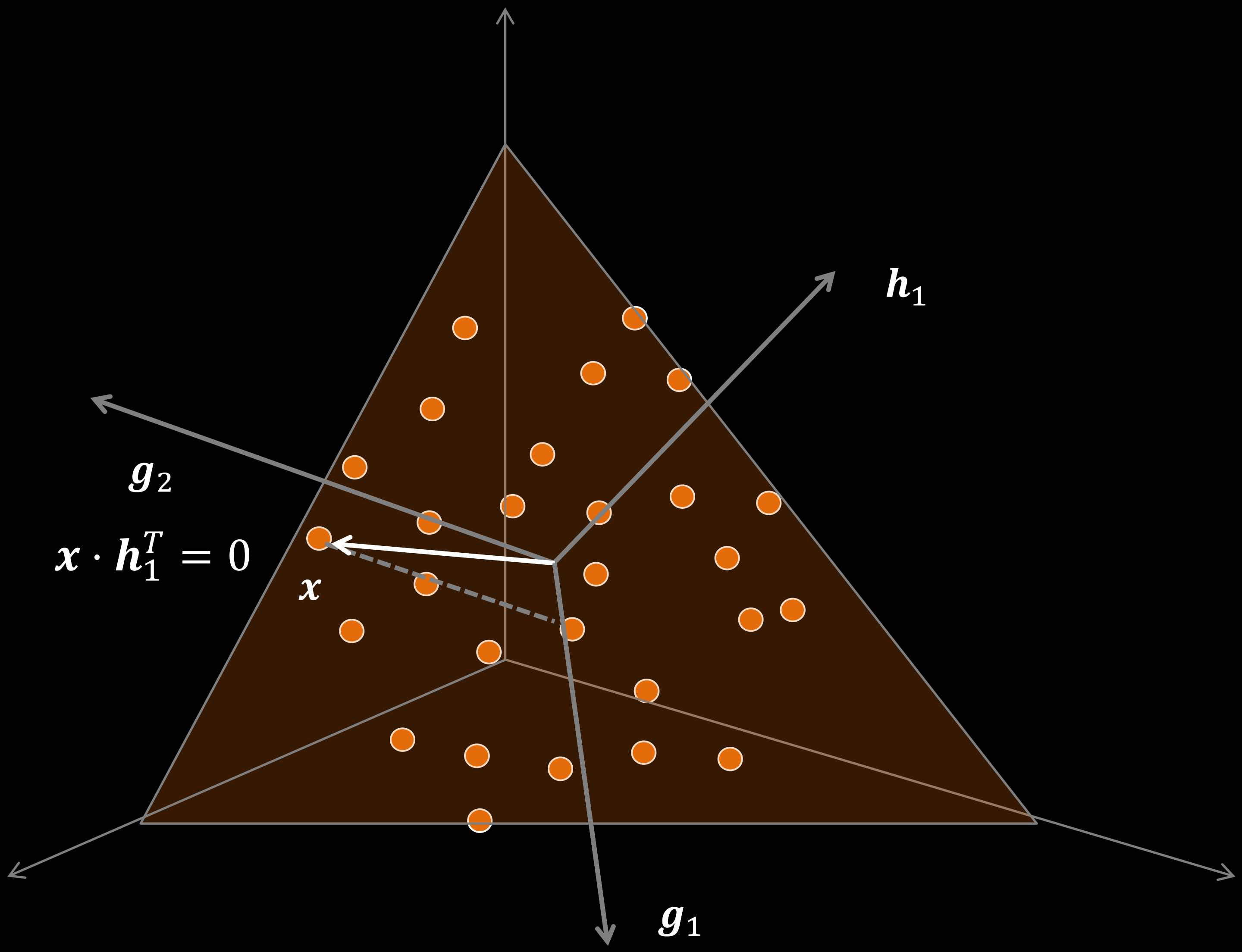
Dimension of a linear block code

$\{\boldsymbol{g}_1, \boldsymbol{g}_2, \dots, \boldsymbol{g}_k\}$ basis for code C

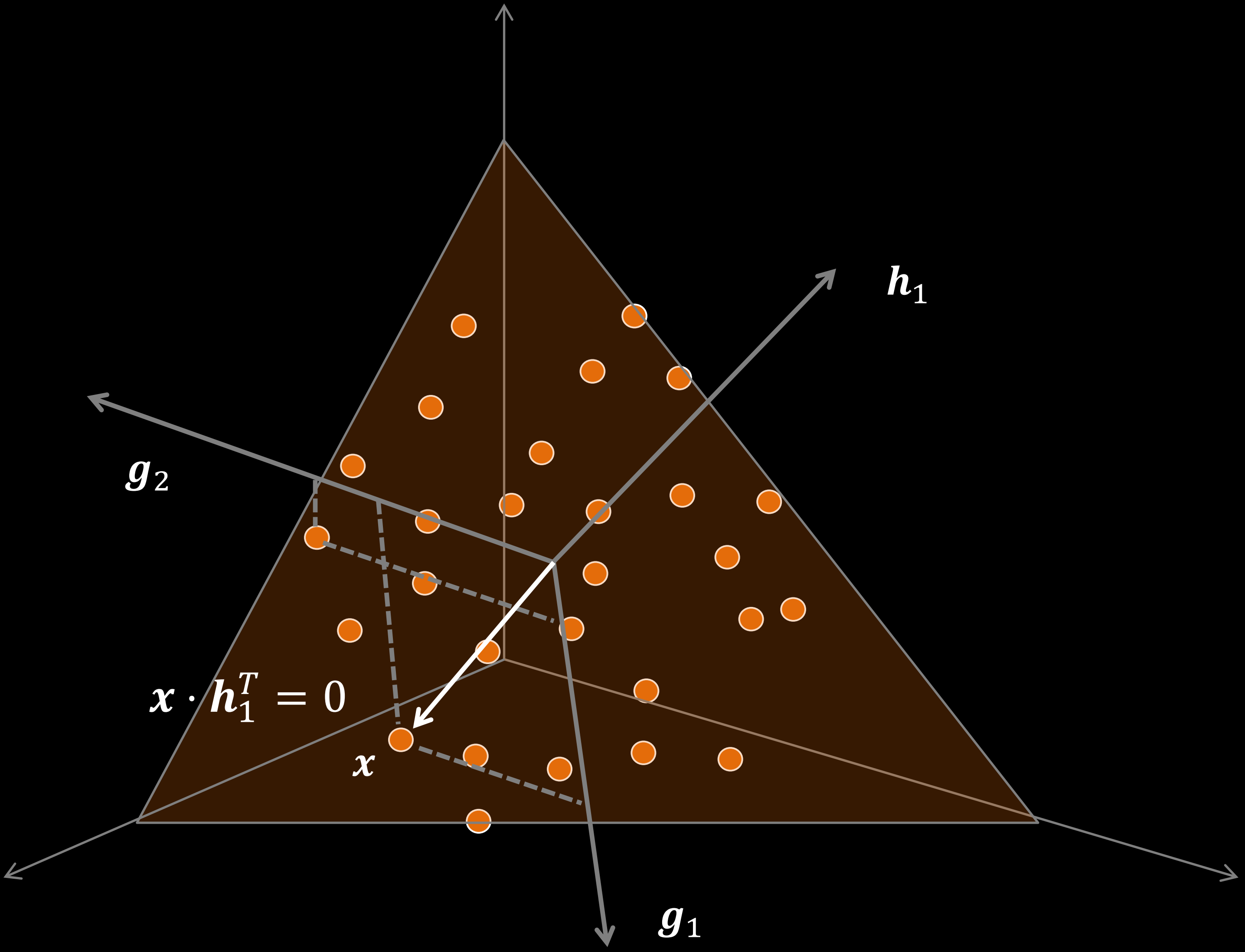
$\{\boldsymbol{h}_1, \boldsymbol{h}_2, \dots, \boldsymbol{h}_{n-k}\}$ the basis of $C^\perp$



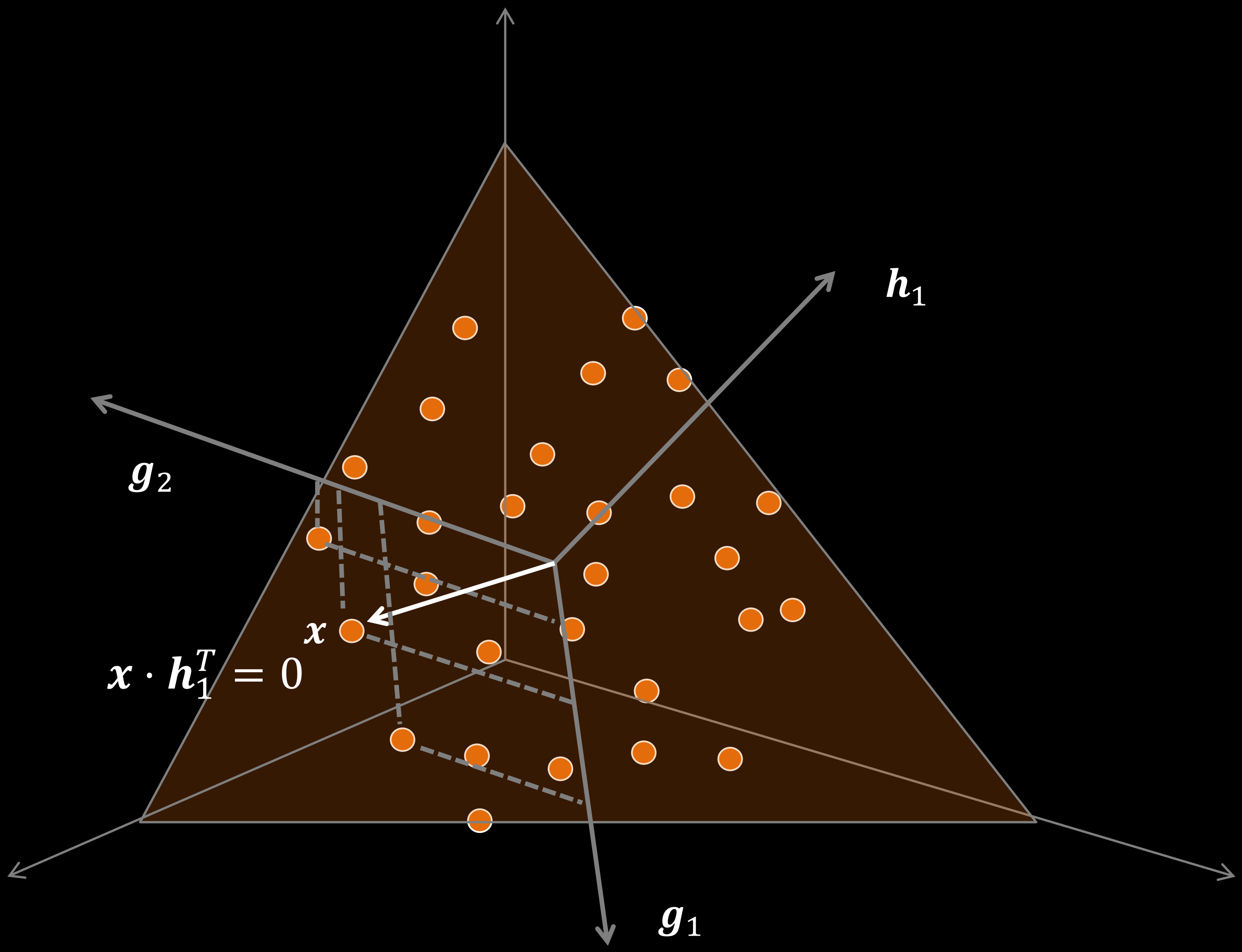
Parity check



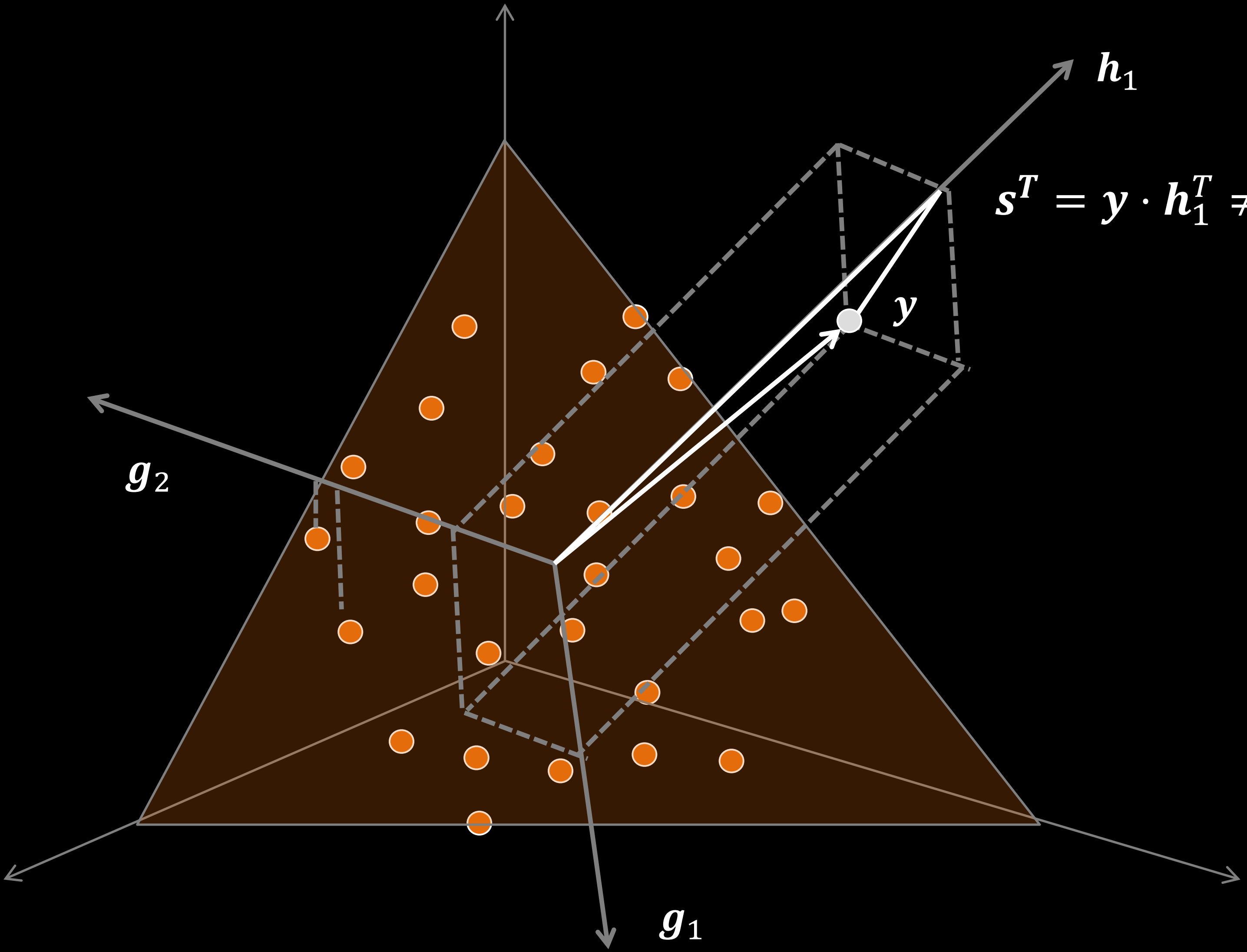
Parity check



Parity check



Syndrome



Poll Question 5

A linear block code with minimum distance *d* can correct any

A. weight - $(d-1)$ errors

B. weight - $\left(\frac{d-1}{2}\right)$ errors

C. weight - $\left(\frac{d}{2}\right)$ errors

D. weight - $\left(\frac{d}{2}+1\right)$ errors

E. I’m not sure

[7,4,3] Hamming code

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

$$x = \begin{bmatrix} m_1 & m_2 & m_3 & m_4 \end{bmatrix} G$$

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

Codewords

0 0 0 0 0 0 0
1 1 0 1 0 0 0
0 1 1 0 1 0 0
1 0 1 1 1 0 0
1 1 1 0 0 1 0
0 0 1 1 0 1 0
1 0 0 0 1 1 0
0 1 0 1 1 1 0
1 0 1 0 0 0 1
0 1 1 1 0 0 1
1 1 0 0 1 0 1
0 0 0 1 1 0 1
0 1 0 0 0 1 1
1 0 0 1 0 1 1
0 0 1 0 1 1 1
1 1 1 1 1 1 1

$m_1 + m_3 + m_4$

$m_1 + m_2 + m_3$

$m_2 + m_3 + m_4$

Example – correcting single errors

- Rewrite H using column vectors $H = [\mathsf{h}_1, \mathsf{h}_2, \ldots, \mathsf{h}_j, \ldots \mathsf{h}_n]$
- Error vector $e = [e_1, e_2, \ldots, e_j, \ldots, e_n]$
- Syndrome $s^{\mathrm{T}} = He^{\mathrm{T}} = [e_1\mathsf{h}_1, e_2\mathsf{h}_2, \ldots, e_j\mathsf{h}_j, \ldots, e_n\mathsf{h}_n]$
- Suppose e contains only one binary 1 at the j-th position, i.e., $e = [0, 0, \ldots, e_j = 1, \ldots 0]$
- Then $s^{\mathrm{T}} = \mathsf{h}_j$
- In order to correct a single error in the codeword, the columns of H must be all different and nonzero.
- The dimensions of H are $(n - k) \times n$, thus the largest code length is $n = 2^{n-k} - 1$.
- Thus, in this case, $k = n - \log_2(n + 1)$.

[7,4,3] Hamming code

$$H = \left[\boxed{\begin{matrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{matrix}}\;\boxed{\begin{matrix} 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \end{matrix}}\right]$$

Each column is a distinct binary vector of length $n - k = 3$

$\Rightarrow$ Corrects 1 error

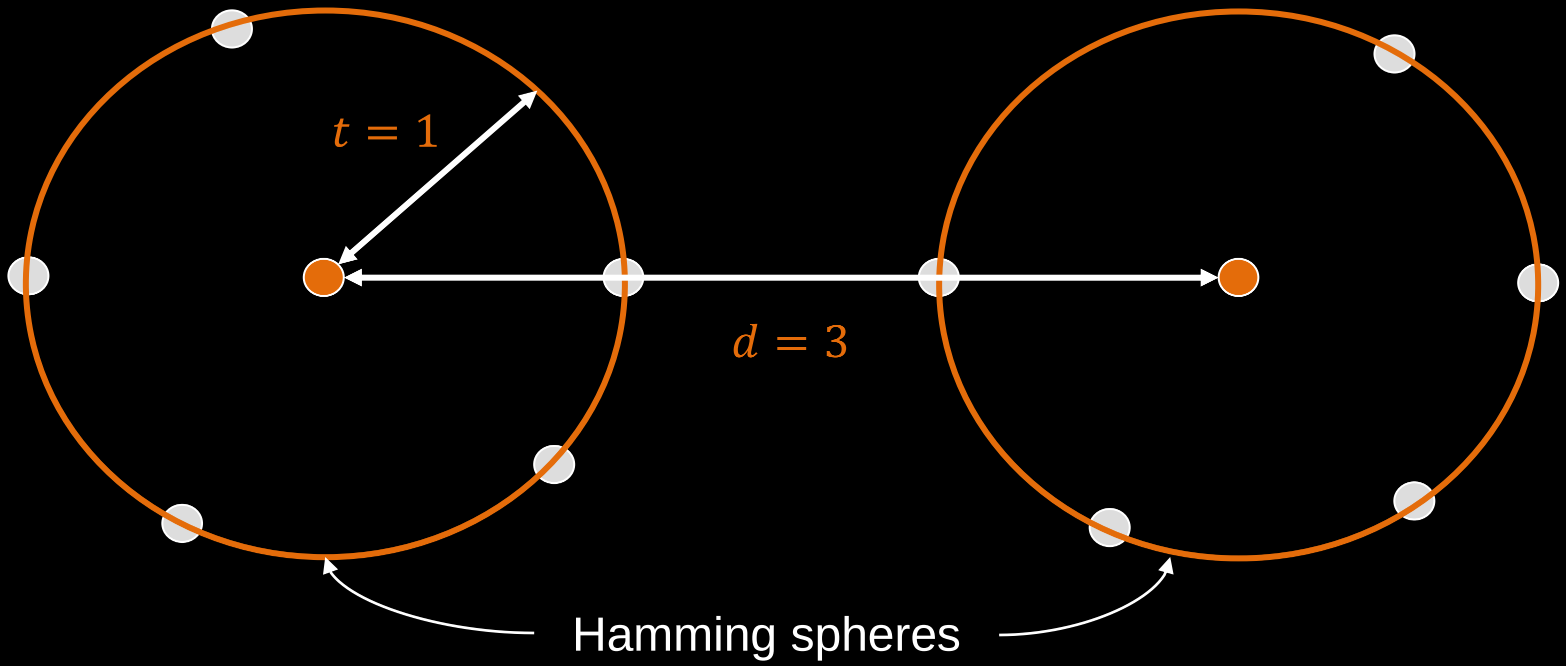
$$G = \left[\boxed{\begin{matrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \end{matrix}}\;\boxed{\begin{matrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{matrix}}\right]$$

$$g_i H^{\mathrm{T}} = 0 \text{ for any row } g_i \text{ of } G$$

$$G H^{\mathrm{T}} = 0$$

[7,4,3] Hamming code

$\binom{n}{t} = 7$ vectors at Hamming distance $t = \frac{d-1}{2} = 1$ from any codeword



Perfect Code: The $2^k = 16$ Hamming spheres cover all the $2^n = 128$ vectors!

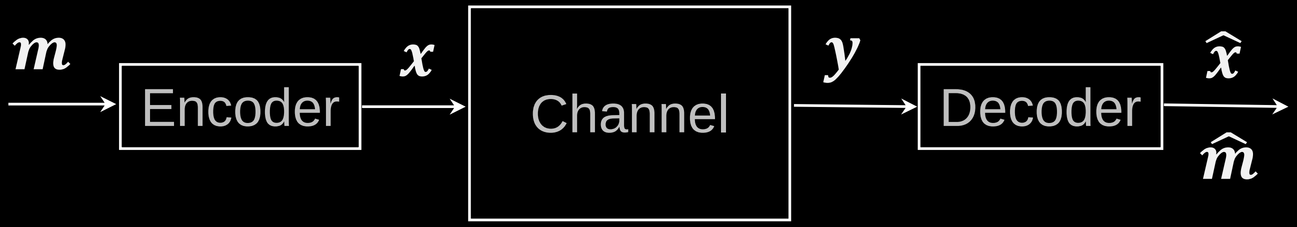
Poll Question 6

The channel introduces random bit flips but in any 10 consecutive bits, it introduces no more than a single bit flip. You choose to use a Hamming code to protect user information bits against bit flips in such channel. The largest number of information bits that can be protected by encoding them using the Hamming code is:

A. 0
B. 1
C. 2
D. 3
E. 4
F. 5
G. 9

H. I’m not sure

Maximum likelihood (ML) decoding



- $d = d(\boldsymbol{x}, \boldsymbol{y})$: the Hamming distance between $\boldsymbol{x}$ and $\boldsymbol{y}$
- For the BSC channel: $P(y|x) = \alpha^d (1-\alpha)^{n-d}$
- ML decoding rule:

$$\hat{x} = \underset{x\in C}{\operatorname{argmax}}\ P(x|y) = \underset{x\in C}{\operatorname{argmax}} \frac{P(x)P(y|x)}{P(y)}$$

- If all codewords are equally likely, then maximize $\log_2 P(y|x)$

ML decoder on BSC channel

- ML decoding rule:

$$\hat{x} = \operatorname*{argmax}_{x\in C} \; \log_2 P(y|x)$$

- For the BSC:

$$\begin{aligned} P(y|x) &= \alpha^d (1-\alpha)^{n-d} \\ \log_2 P(y|x) &= d\log_2 \alpha + (n-d)\log_2(1-\alpha) \\ &= d\log_2 \frac{\alpha}{1-\alpha} + n\log_2(1-\alpha) \end{aligned}$$

negative for $\alpha < 1/2$ independent of d

- Hence, $\hat{\boldsymbol{x}}$ is the codeword closest to $\boldsymbol{y}$:

$$\hat{x} = \operatorname*{argmin}_{x\in\mathcal{C}} d(x,y)$$

Example [6,3,3] code

$$G = \begin{bmatrix} 1 & & & 1 & & 1 \\ & 1 & & 1 & 1 & \\ & & 1 & & 1 & 1 \end{bmatrix} \qquad H = \begin{bmatrix} 1 & 1 & & 1 & & \\ & 1 & 1 & & 1 & \\ 1 & & 1 & & & 1 \end{bmatrix}$$

$$C = \left\{ \begin{array}{l} 000000, \\ 001011, \\ 010110, \\ 100101, \\ 011101, \\ 101110, \\ 110011, \\ 111000 \end{array} \right\}$$

Standard array decoding

ML decoding: $\hat{x} = \underset{x \in \mathcal{C}}{\operatorname{argmin}}\, d(x, y)$

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

2^k

2^{n-k}

000 000	001 011	010 110	100 101	011 101	101 110	110 011	111 000	000
000 001	001 010	010 111	100 100	011 100	101 111	110 010	111 001	001
000 010	001 001	010 100	100 111	011 111	101 100	110 001	111 010	010
000 100	001 111	010 010	100 001	011 001	101 010	110 111	111 100	100
001 000	000 011	011 110	101 101	010 101	100 110	111 011	110 000	011
010 000	011 011	000 110	110 101	001 101	111 110	100 011	101 000	110
100 000	101 011	110 110	000 101	111 101	001 110	010 011	011 000	101

Standard array decoding

ML decoding: $\hat{x} = \underset{x \in \mathcal{C}}{\operatorname{argmin}}\, d(x, y = 110\,101)$

2^k

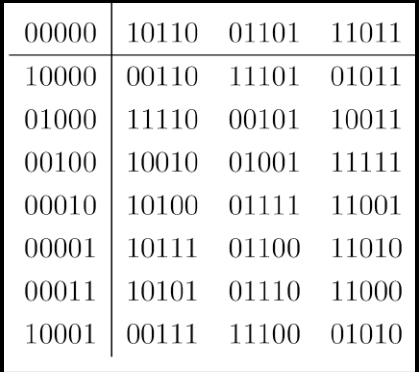
2^{n-k}

000 000	001 011	010 110	$\boxed{\text{100 101}}$	011 101	101 110	110 011	111 000	000
000 001	001 010	010 111	100 100	011 100	101 111	110 010	111 001	001
000 010	001 001	010 100	100 111	011 111	101 100	110 001	111 010	010
000 100	001 111	010 010	100 001	011 001	101 010	110 111	111 100	100
001 000	000 011	011 110	101 101	010 101	100 110	111 011	110 000	011
010 000	011 011	000 110	**110 101**	001 101	111 110	100 011	101 000	110
100 000	101 011	110 110	000 101	111 101	001 110	010 011	011 000	101

Complexity scales exponentially!!

Poll Question 7

The standard array of a linear block code is given below.



00000	10110	01101	11011
10000	00110	11101	01011
01000	11110	00101	10011
00100	10010	01001	11111
00010	10100	01111	11001
00001	10111	01100	11010
00011	10101	01110	11000
10001	00111	11100	01010

The word received from the channel is $\boldsymbol{y} = [0\ 1\ 1\ 0\ 0]$, and the standard array decoder is used to estimate the transmitted codeword $\boldsymbol{x}$. The estimated codeword is:

A. The standard array decoder cannot correct this error pattern

B. $\boldsymbol{x} = [0\ 0\ 0\ 0\ 1]$

C. $\boldsymbol{x} = [0\ 0\ 1\ 0\ 1]$

D. $\boldsymbol{x} = [0\ 1\ 1\ 0\ 0]$

E. $\boldsymbol{x} = [0\ 1\ 1\ 0\ 1]$

F. $\boldsymbol{x} = [0\ 1\ 1\ 0\ 0]$

G. I’m not sure

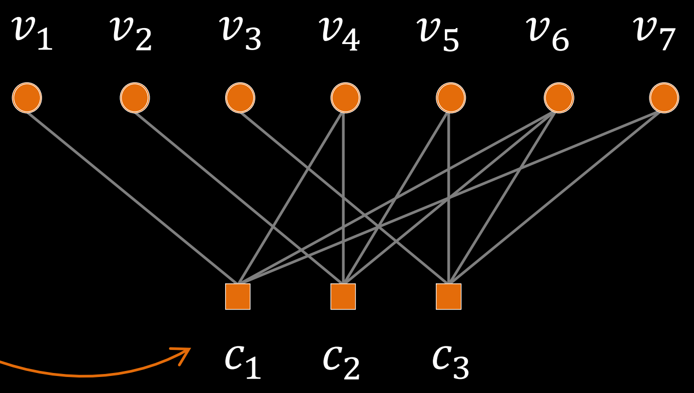
Codes on Graphs
and
Iterative Decoding

Graphical model for a linear block code

$$
\begin{aligned}
c_1&: & v_1 + v_4 + v_6 + v_7 &= 0\\
c_2&: & v_2 + v_4 + v_5 + v_6 &= 0\\
c_3&: & v_3 + v_5 + v_6 + v_7 &= 0
\end{aligned}
$$

Checks

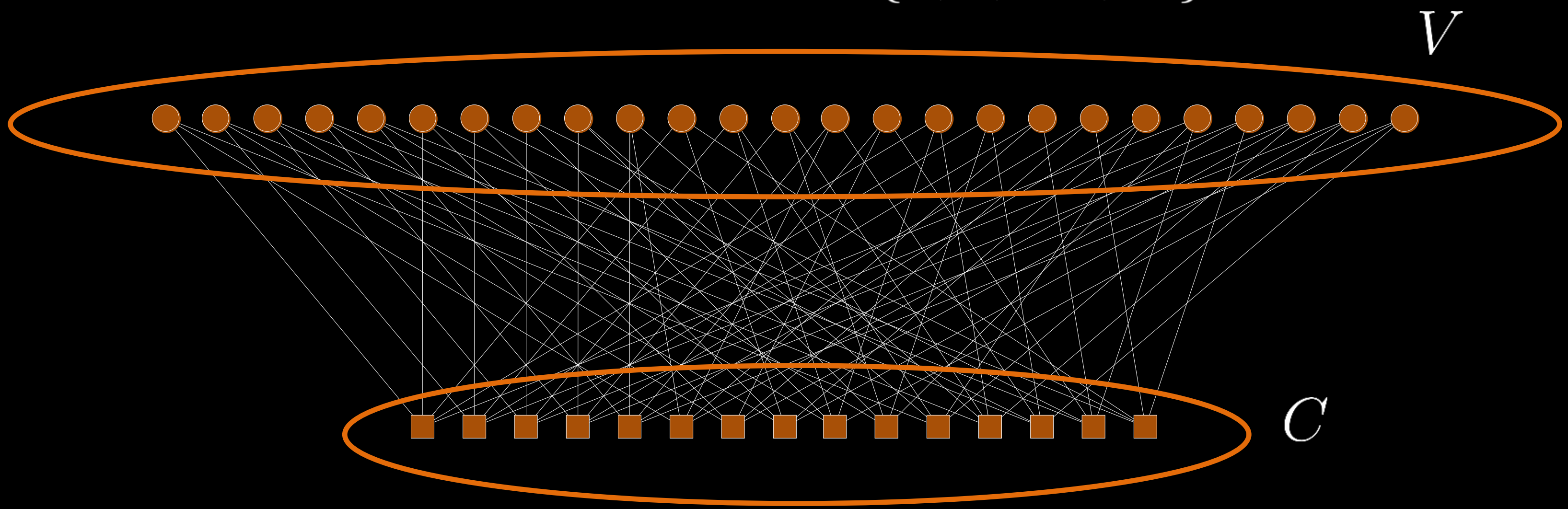
Variables



Factor graph
(or)
Tanner graph

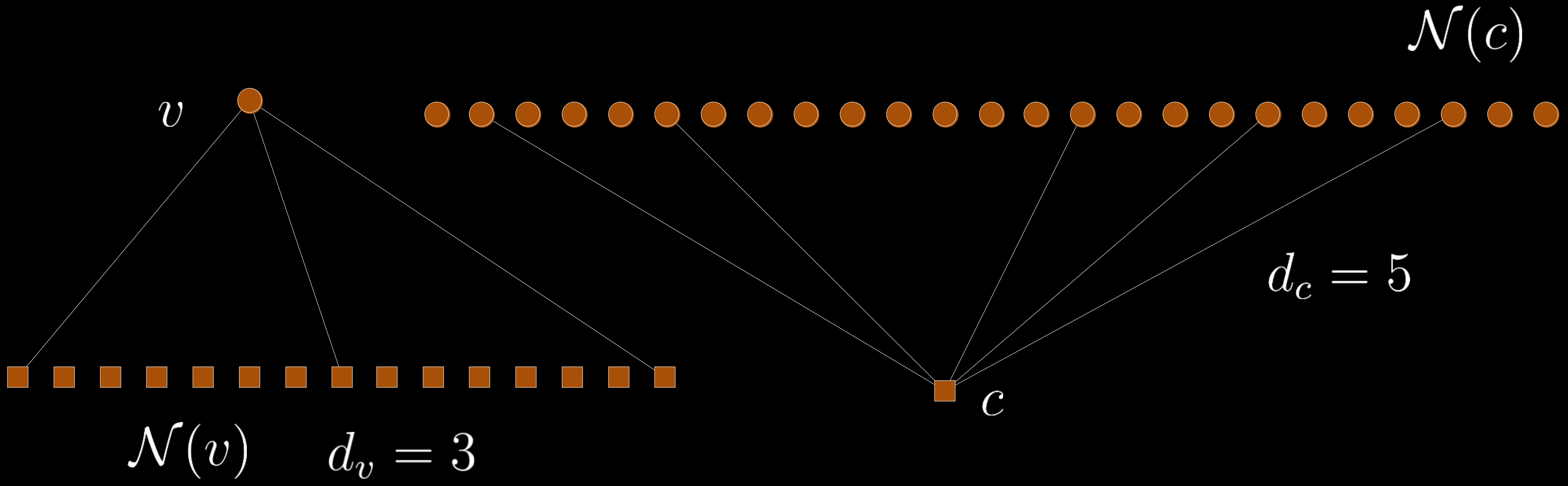
Low-density parity-check (LDPC) codes

- Linear block codes defined by sparse bipartite graphs
- The *Tanner* graph of an LDPC code C is a bipartite graph G with two sets of nodes:
  - the set of variable nodes $V = \{1, 2, \ldots, n\}$
  - and the set of check nodes $C = \{1, 2, \ldots, m\}$



Definitions

- The check nodes (resp. variable nodes) connected to a variable node (resp. check node) are its "*neighbors*".
- The set of neighbors of a node u is denoted by $\mathcal{N}(u)$
- The degree d_u of a node u is the size of $\mathcal{N}(u)$

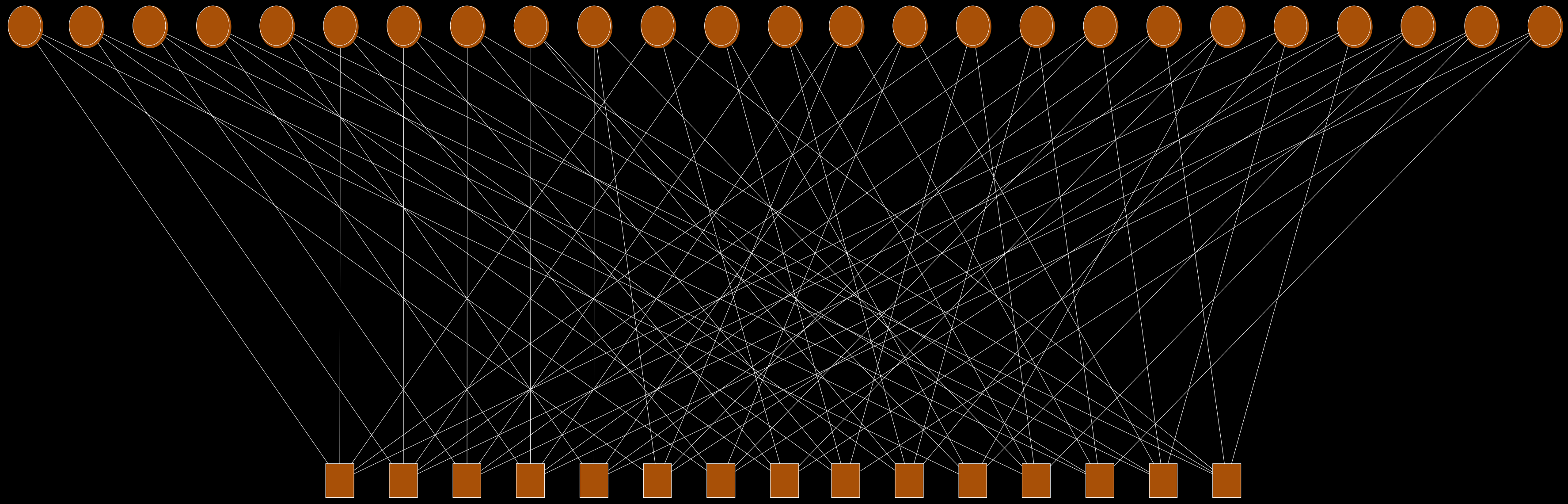


Definitions

- A vector $\mathbf{v} = [v_1, v_2, \ldots, v_n]$ is a codeword if and only if for each check node, the modulo two sum of its neighbors (i.e., respective bits of the vector) is zero
- An (n, γ, ρ) regular LDPC code has a Tanner graph with n variable nodes each of degree γ and $m = n\gamma/\rho$ check nodes each of degree ρ
- This code has length n and rate $r = \frac{k}{n} \geq 1 - \frac{\gamma}{\rho}$

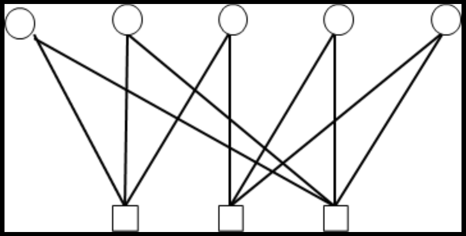
- The Tanner graph is not uniquely defined by the code
- Each parity-check matrix produces one Tanner graph

A regular $(n = 25, \gamma = 3, \rho = 5)$ LDPC code



Poll Question 8

The parity check matrix that corresponds to the following Tanner graph is



$$A.\quad H = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 \end{bmatrix}$$

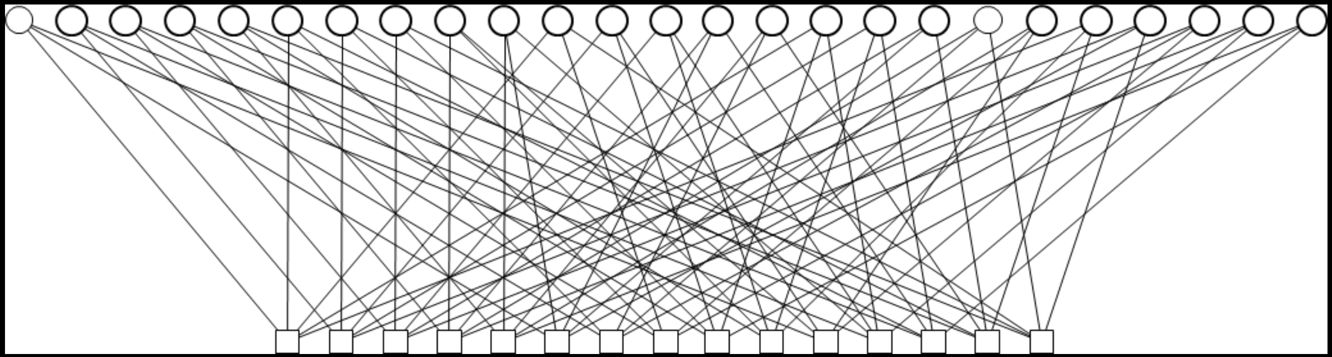
$$B.\quad H = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 \end{bmatrix}$$

$$C.\quad H = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$D.\quad H = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 \end{bmatrix}$$

Poll Question 9

For the code given by the Tanner graph below, the following statement is false:



A. $n = 25$ and $\rho = 5$

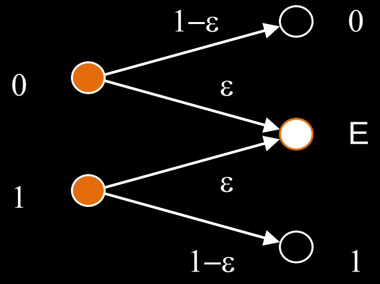
B. The check degree is three

C. $n = 25$ and the Tanner graph is bipartite

D. It is a regular LDPC code

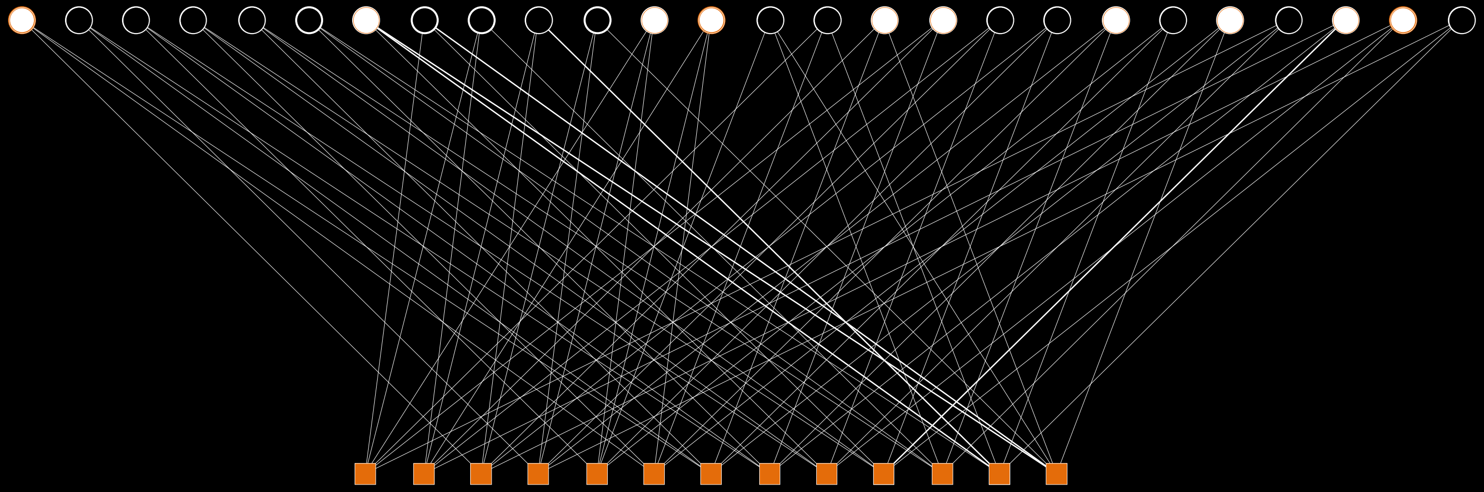
Iterative decoders for BEC

Iterative decoding on BEC



● erased bit

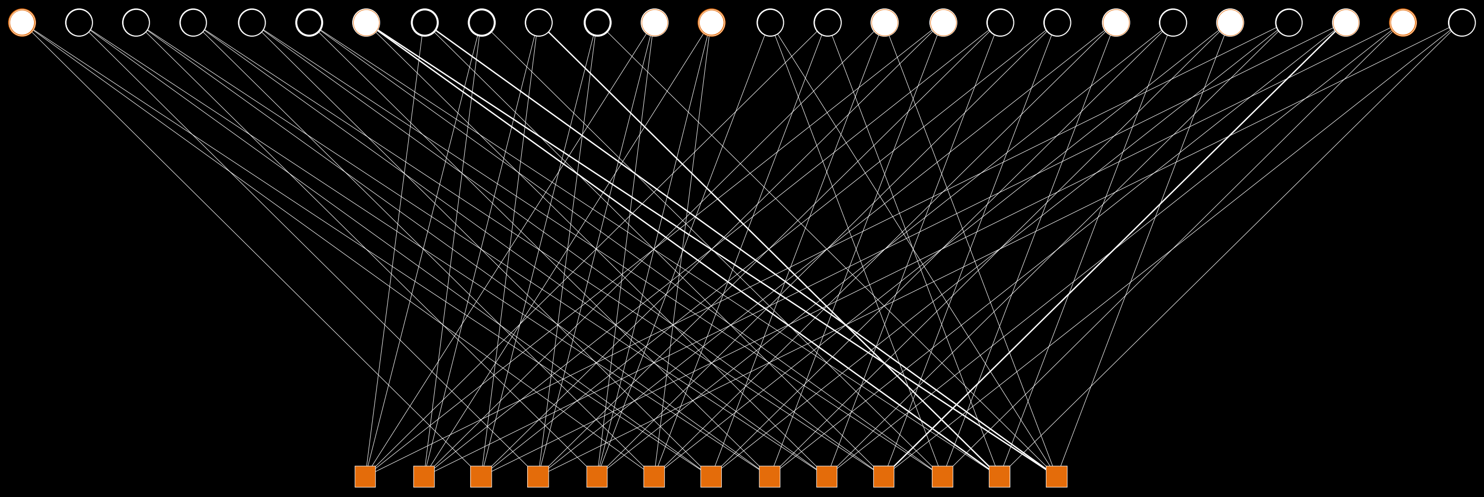
○ correct bit



BEC decoding simulation

● erased bit

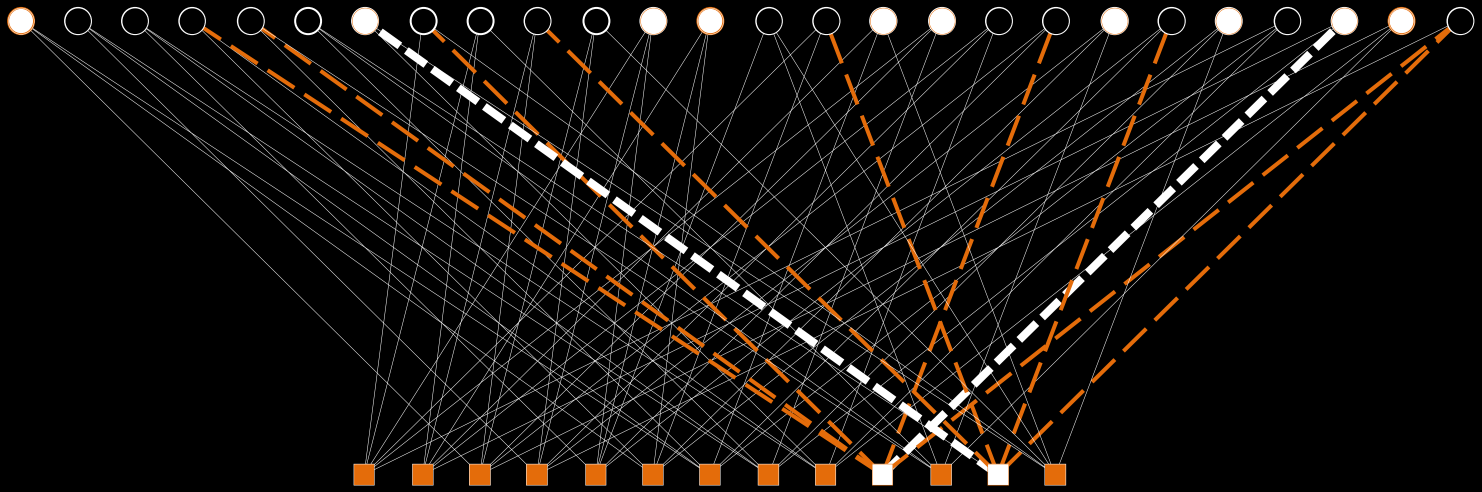
○ correct bit



BEC decoding simulation

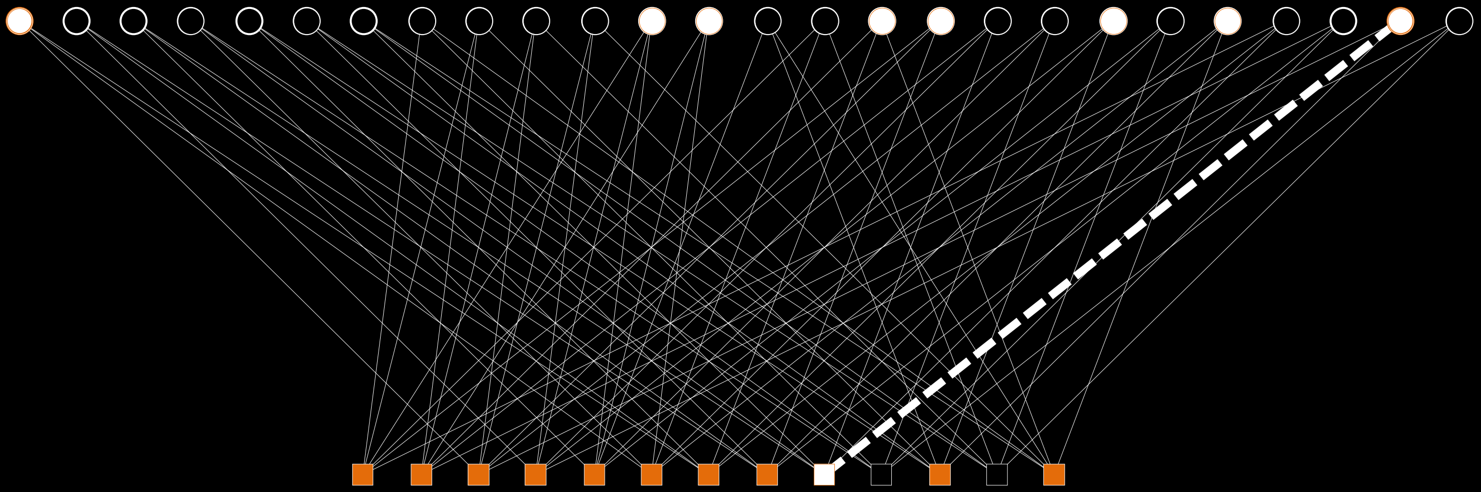
● erased bit

○ correct bit



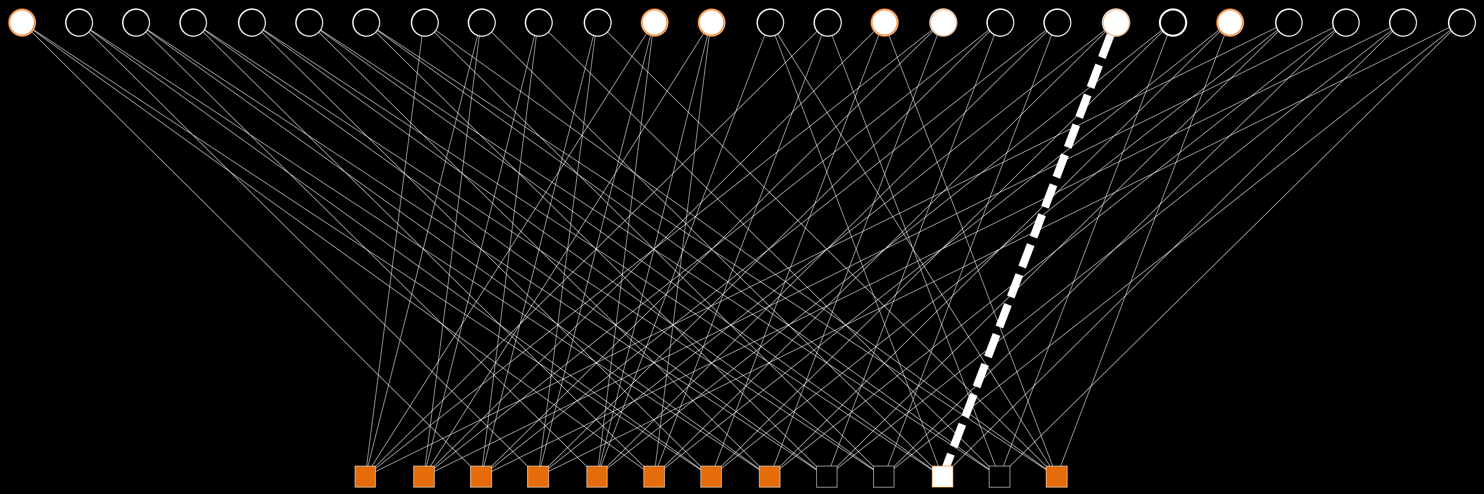
- □ a check involving a single erased bit
- 🟧 other check

BEC simulation - 1



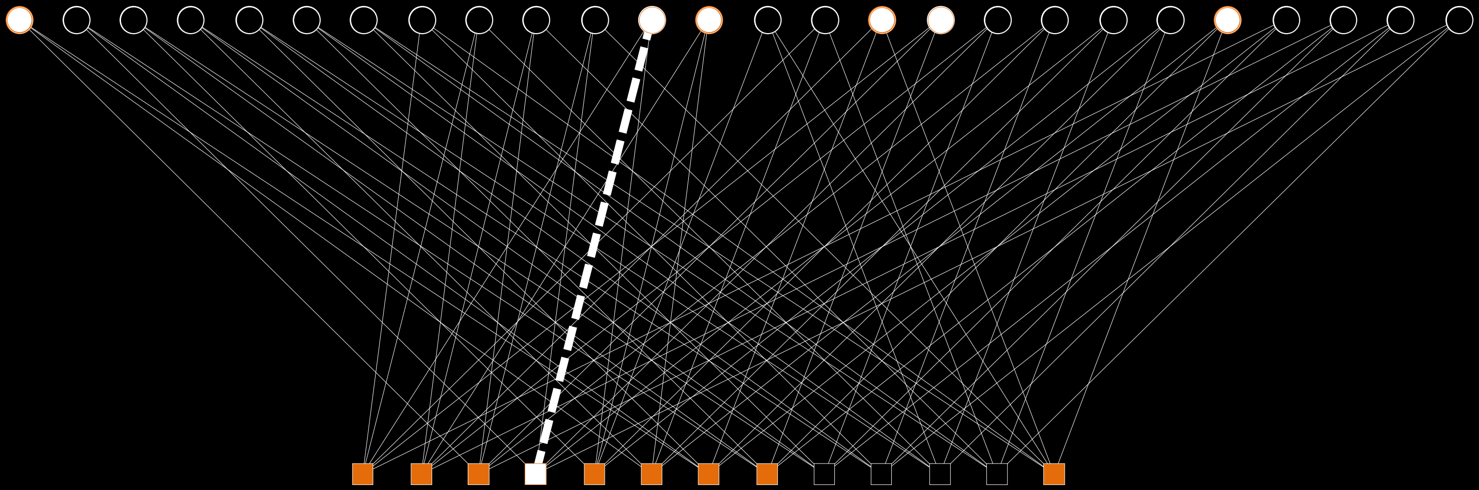
□ a check satisfied after correction

BEC simulation - 2



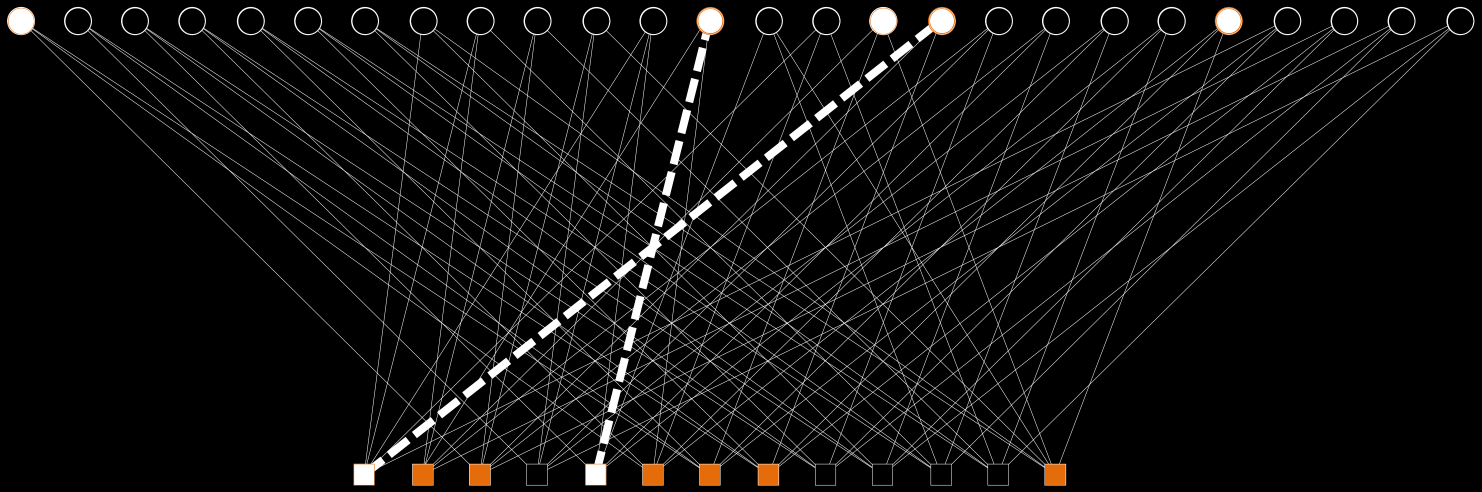
□ a check satisfied after correction

BEC simulation - 3



□ a check satisfied after correction

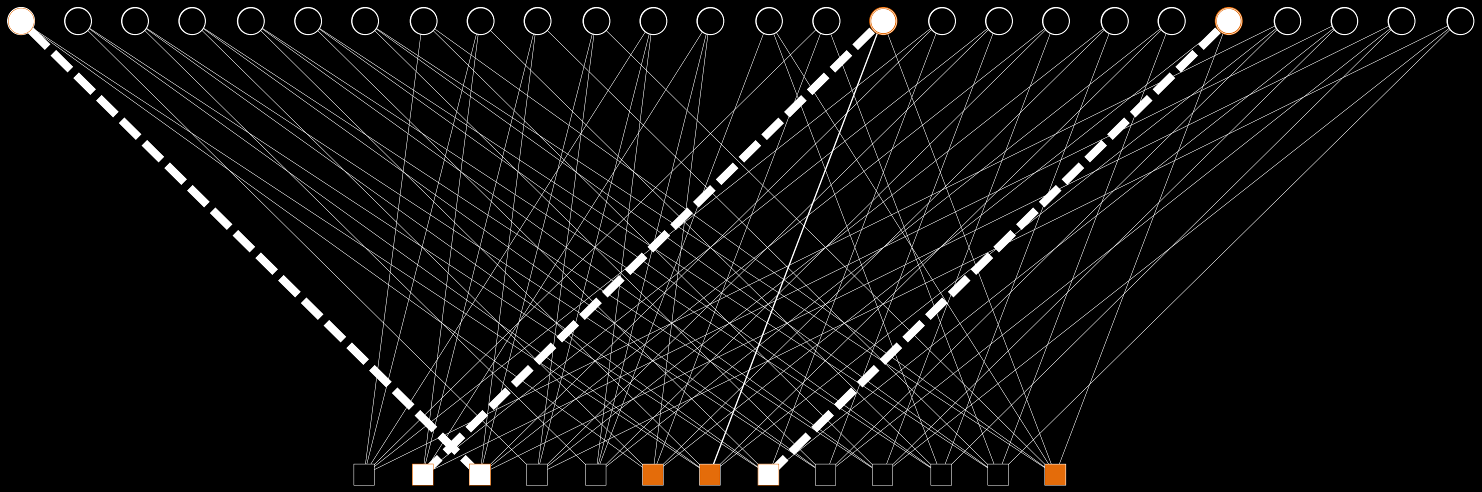
BEC simulation - 4



□ a check satisfied after correction

Center for
Quantum
Networks
NSF-ERC

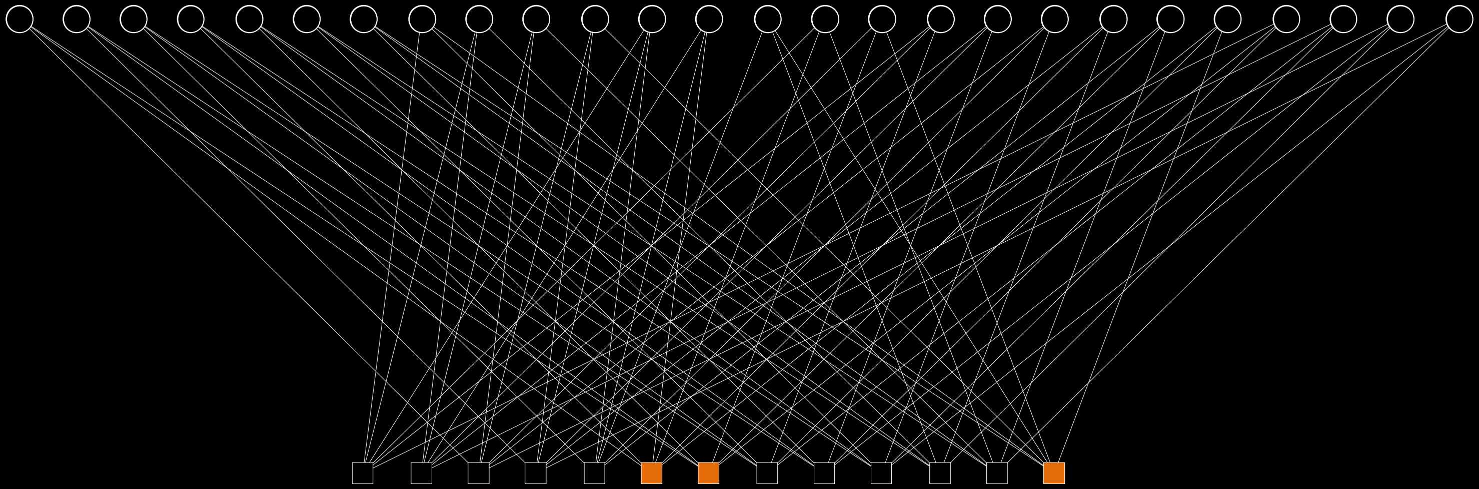
BEC simulation - 5



□ a check satisfied after correction

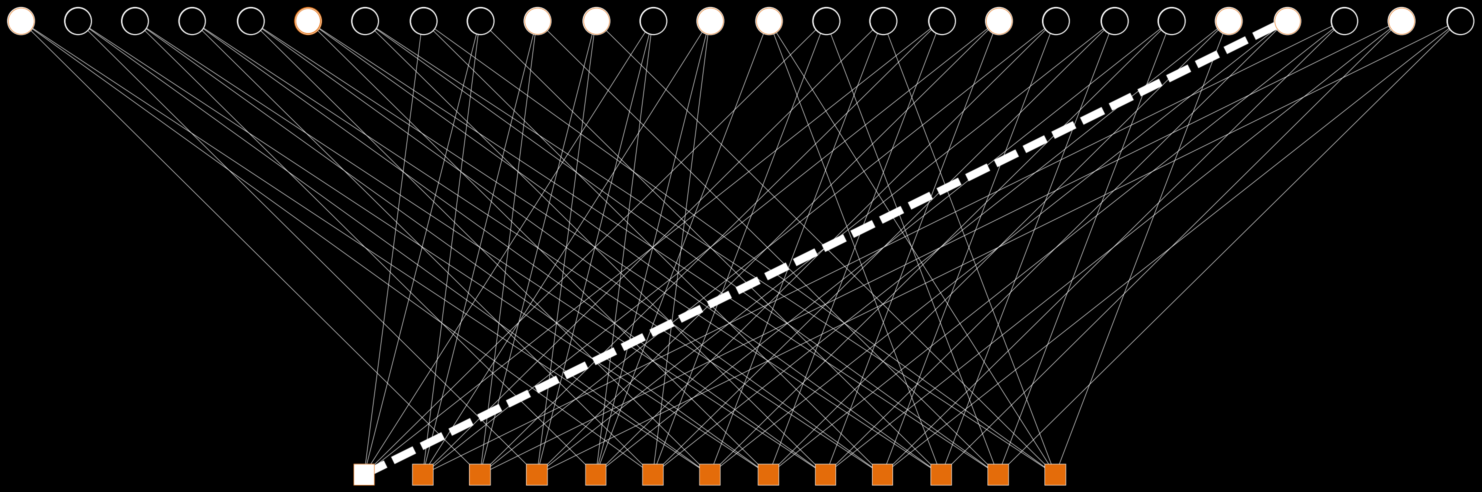
THE UNIVERSITY OF ARIZONA.
TUCSON ARIZONA

BEC simulation - 6

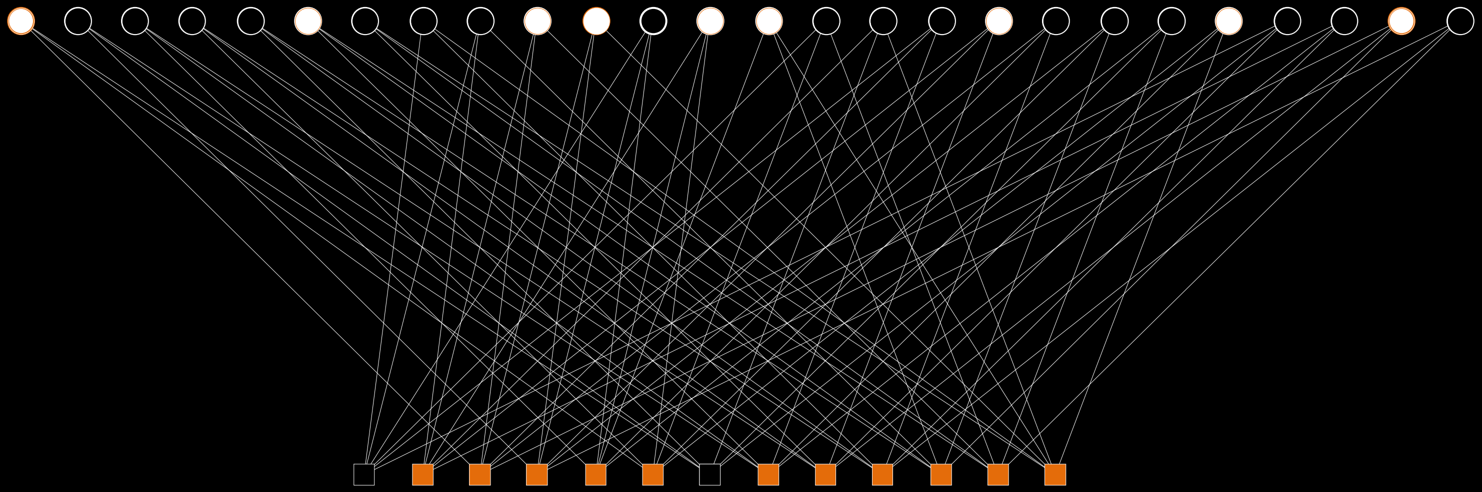


Success !

Another example BEC simulation - 1

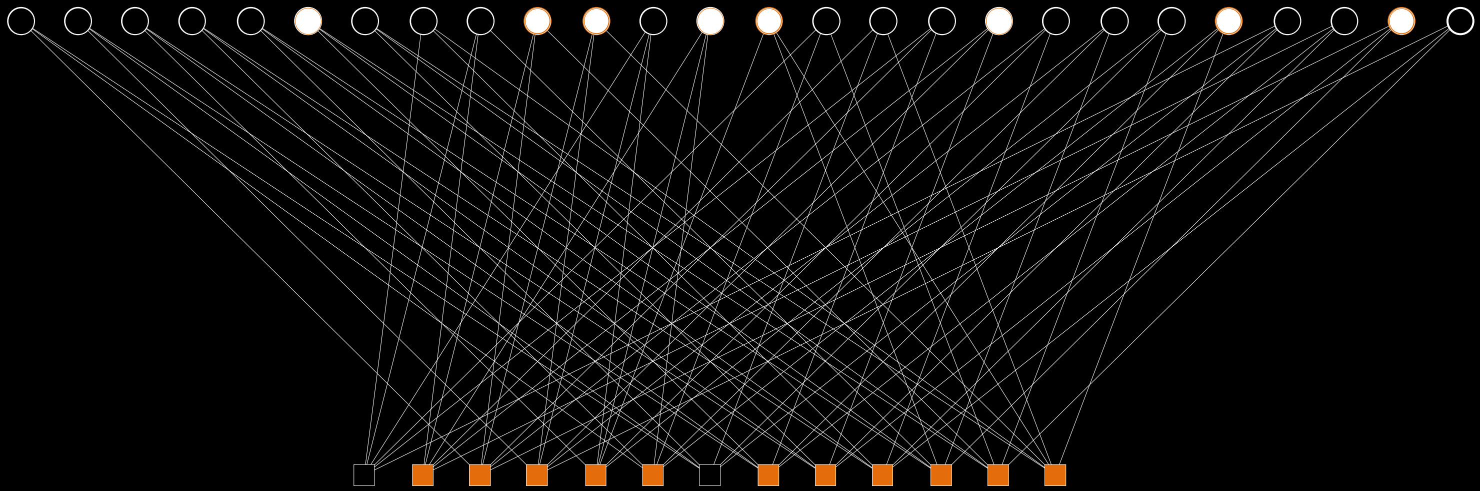


Another example BEC simulation - 2



Center for
Quantum
Networks
NSF-ERC

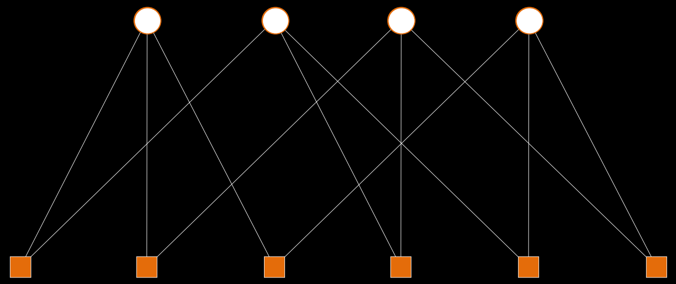
BEC simulation - final



Stuck !

Decoding failures

- A BEC iterative decoder fails to converge to a codeword (correct or wrong) if at any iteration there is no check node connected to at most one erased variable node.



- Graph induced by a subset of check nodes each connected to at least two erased variables is a stopping set.

Gallager A/B algorithm

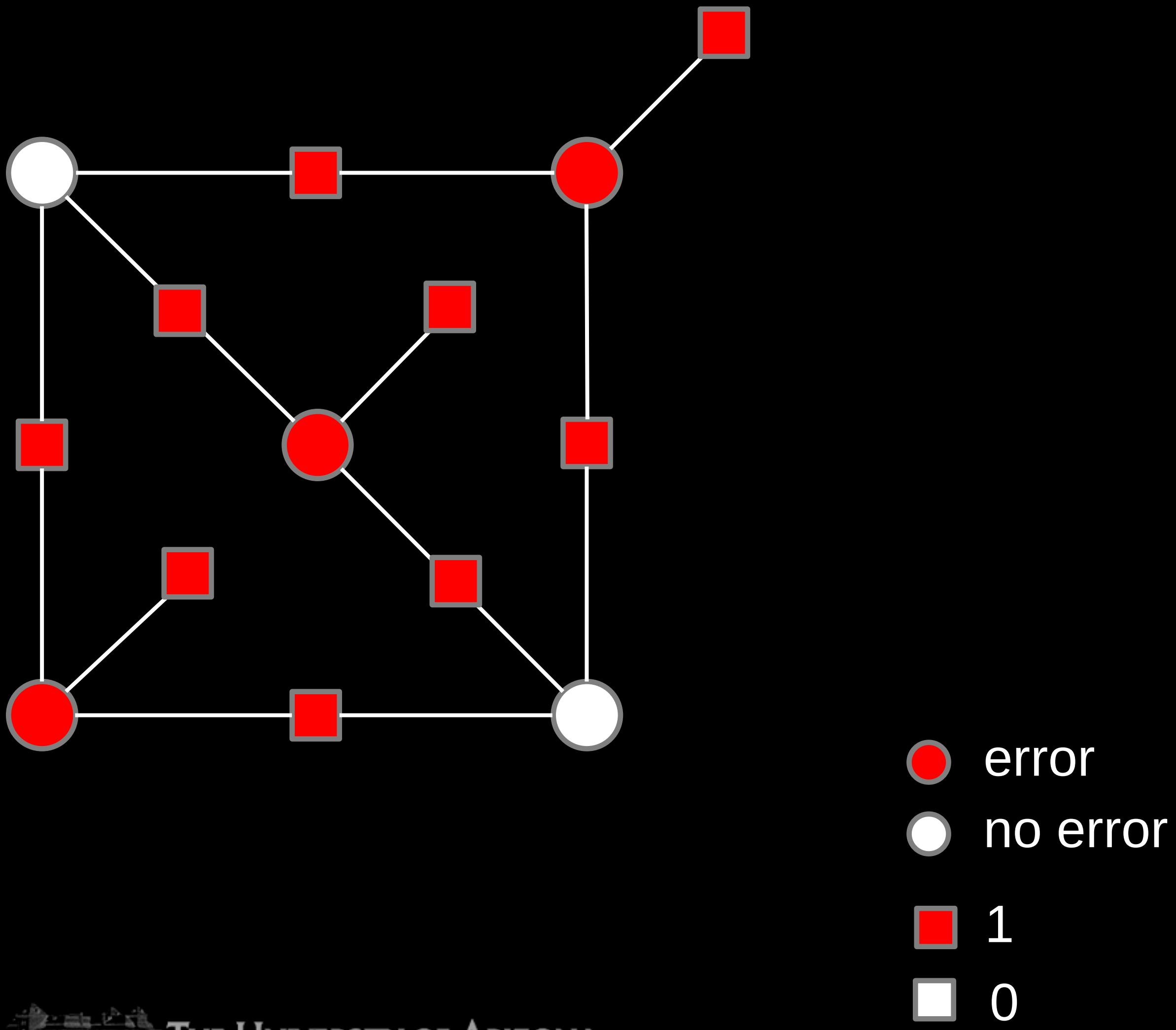
- The Gallager A/B algorithms are hard decision decoding algorithms in which all the messages are binary
- $|\varpi_{*\to i} = m|$ number of incoming messages to i which are equal to $m \in \{0,1\}$. Associated with every decoding round k and variable degree d_i is a threshold b_{k,d_i}.
- The Gallager B algorithm is defined as follows:

$$\begin{aligned}
\omega_{i\to\alpha}^{(0)} &= y_i \\
\varpi_{\alpha\to i}^{(k)} &= \left(\sum_{j\in\mathcal{N}(\alpha)\backslash i} \omega_{j\to\alpha}^{(k-1)}\right) \bmod 2 \\
\omega_{i\to\alpha}^{(k)} &= \begin{cases} 1, & \text{if } |\varpi_{*\backslash\alpha\to i}^{(k)} = 1| \geq b_{k,d_i} \\ 0, & \text{if } |\varpi_{*\backslash\alpha\to i}^{(k)} = 0| \geq b_{k,d_i} \\ y_i, & \text{otherwise} \end{cases}
\end{aligned}$$

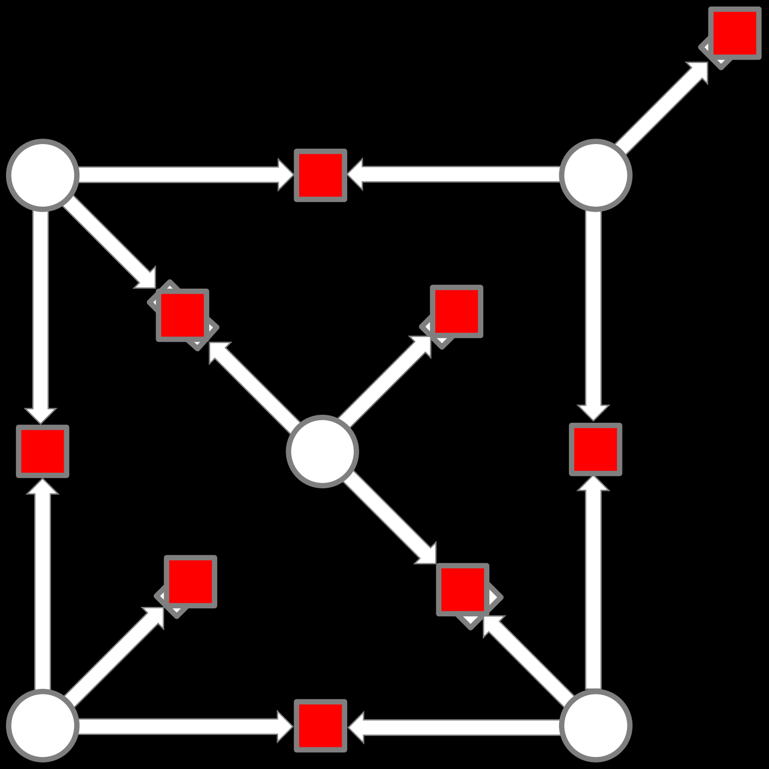
Gallager A

$$b_{k,d_i} = d_i - 1$$

Iterations of Gallager B



Iterations of Gallager B

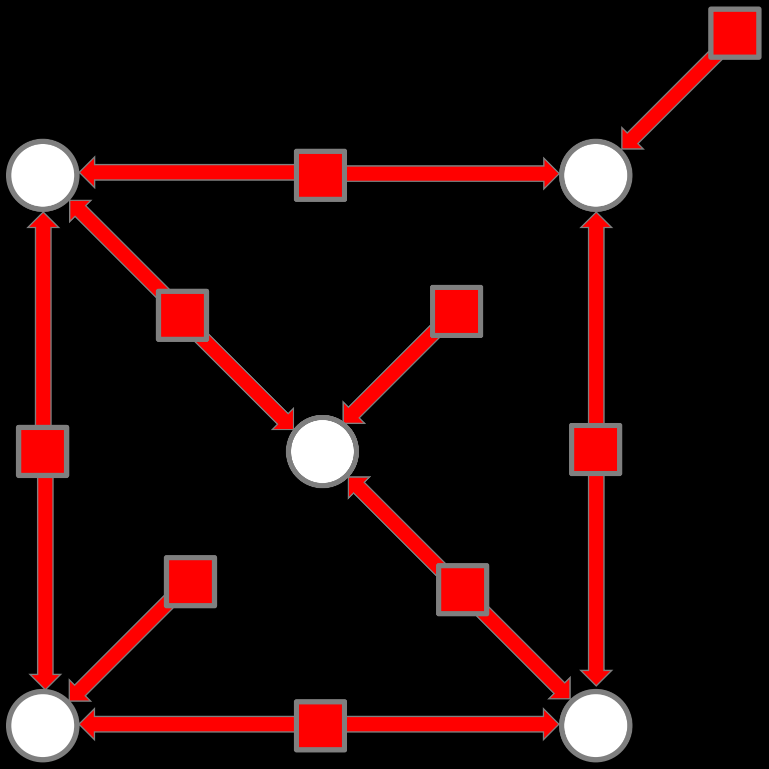


iteration 1 – initialization
all variables send zero

⬜ 0

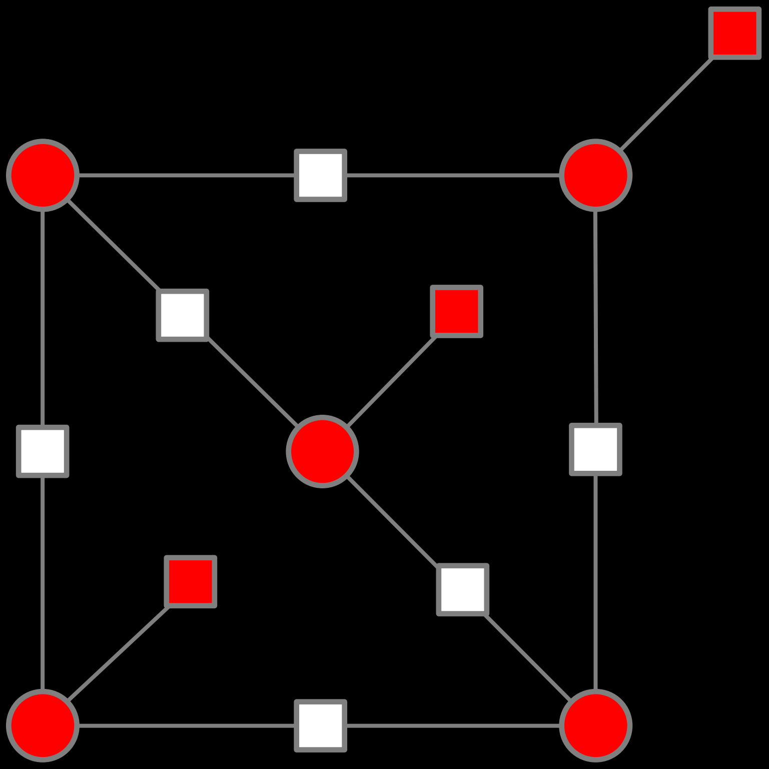
🟥 1

Iterations of Gallager B



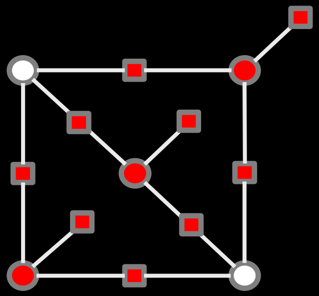
iteration 1 – the second half

Iterations of Gallager B

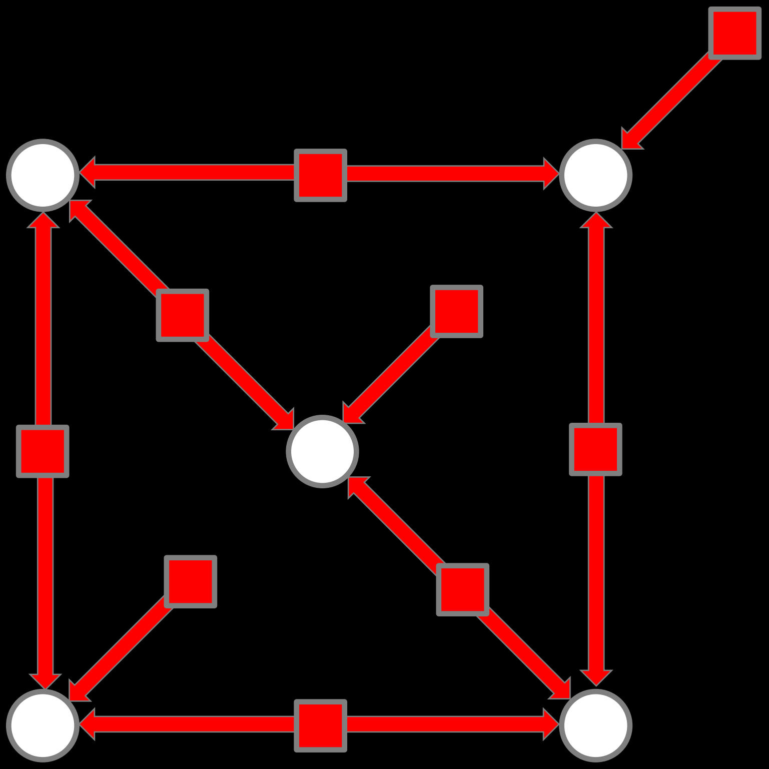


Iteration 1 - decison

syndrome mismatch

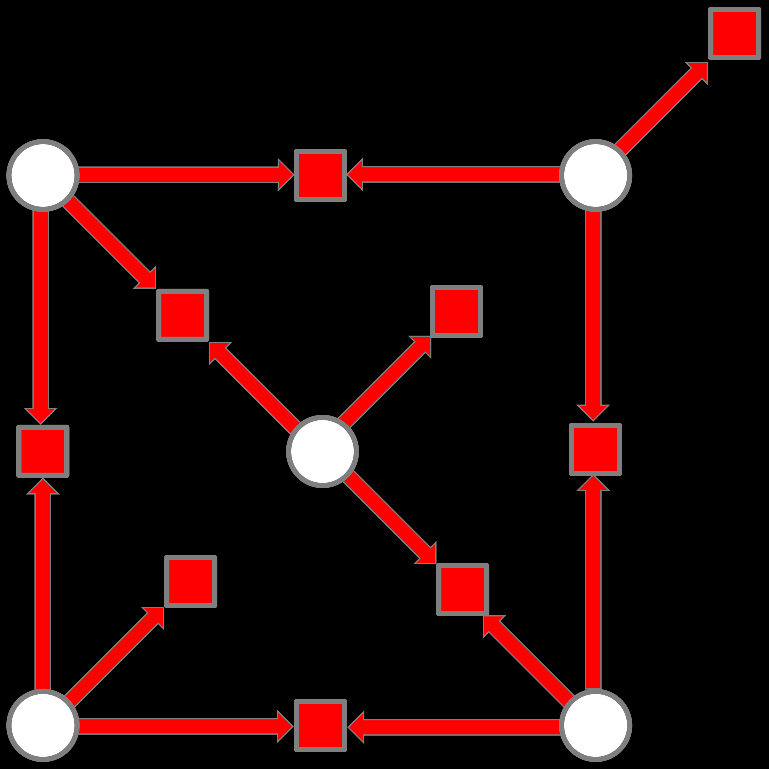


Iterations of Gallager B



recall what messages were sent to variable nodes

Iterations of Gallager B

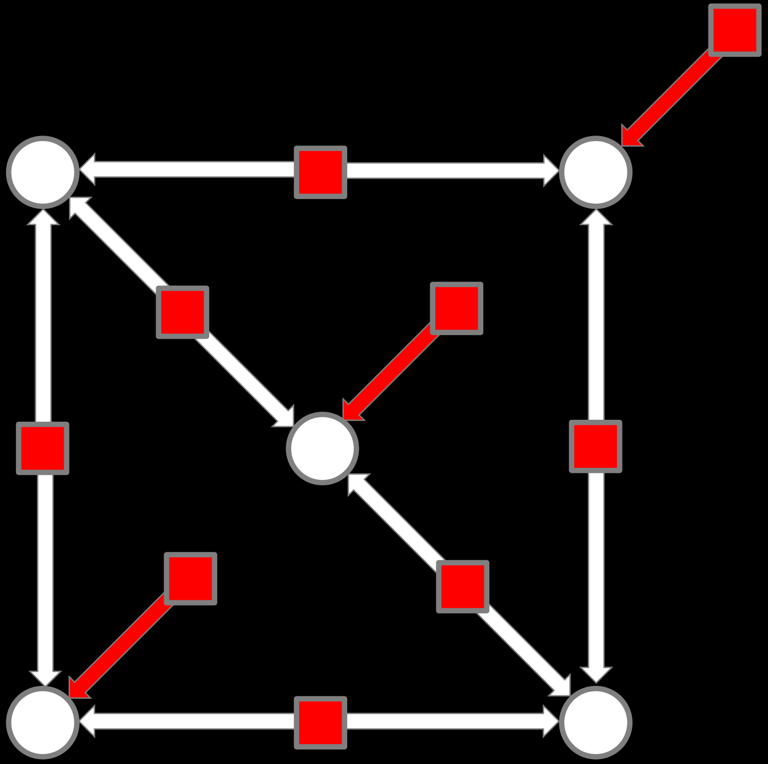


iteration 2 – first half
variables send the majority
of incoming messages

⇨ 0

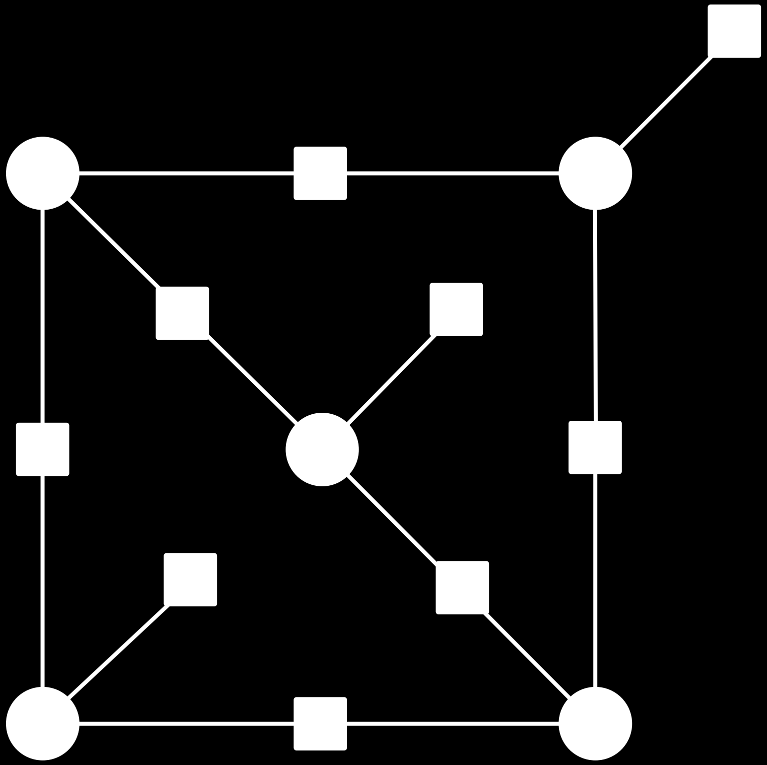
➡ 1

Iterations of Gallager B



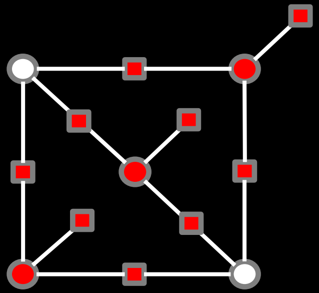
iteration 2 – second half

Iterations of Gallager B

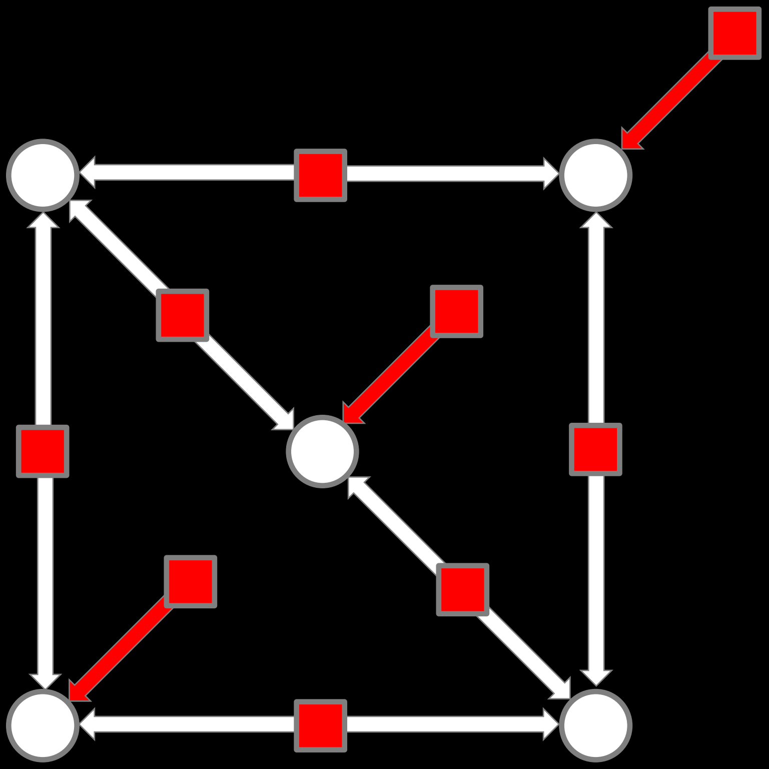


iteration 2 - decision

syndrome mismatch

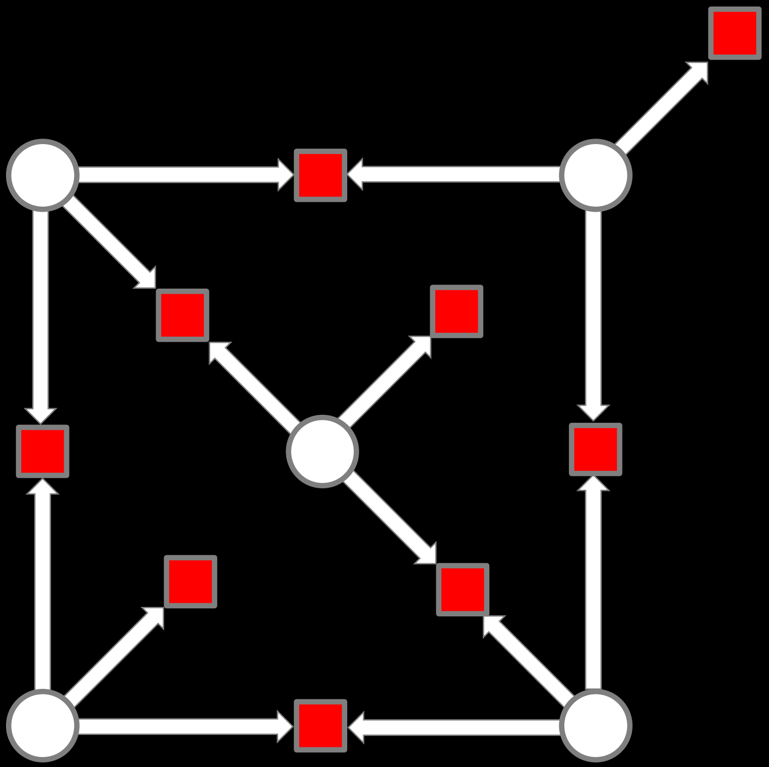


Iterations of Gallager B



messages sent to variable nodes in previous iteration

Iterations of Gallager B



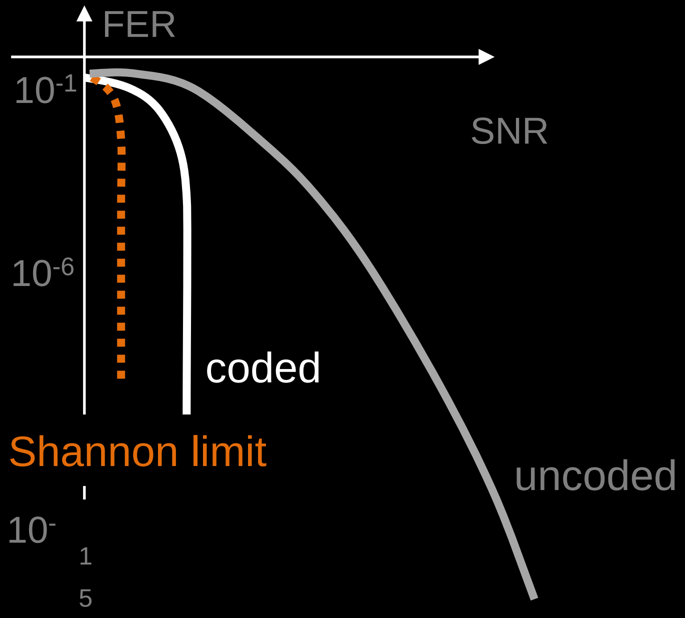
iteration 3 – first half

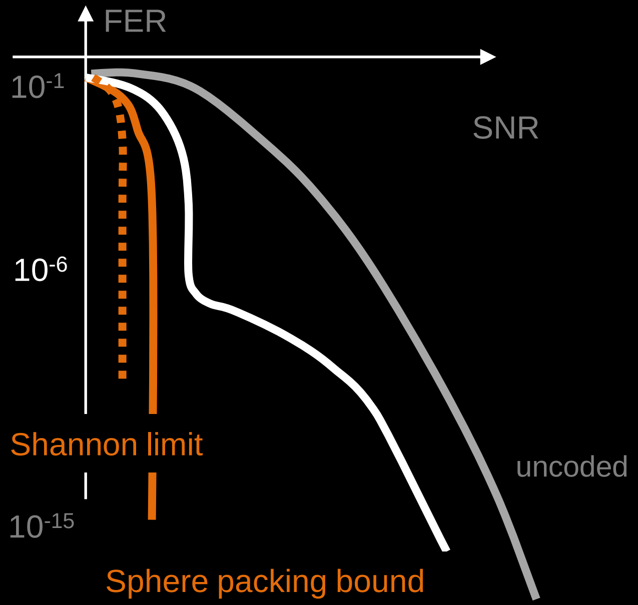
as when we started

⇨ 0

➡ 1

Error floor


FER
10-1
SNR
10-6
coded
Shannon limit
uncoded
10-15


FER
10-1
SNR
10-6
Shannon limit
uncoded
10-15
Sphere packing bound

Poll Question 10

The Gallager-B decoder on the BSC channel with cross over probability α operates by sending messages between variable and check node processing units. After receiving all three messages from its neighboring checks, assuming that the channel value is 0, the variable node processing unit of the variable shown in the picture below will send the following message to the check node processing unit of the remaining check:

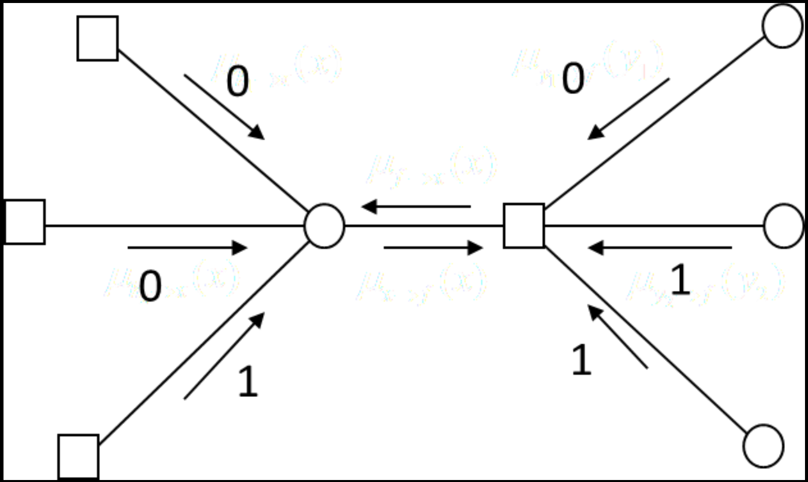
A. $\log\frac{1-\alpha}{\alpha}$

B. $\log\frac{1-\alpha}{\alpha}0$

C. 0

D. 1

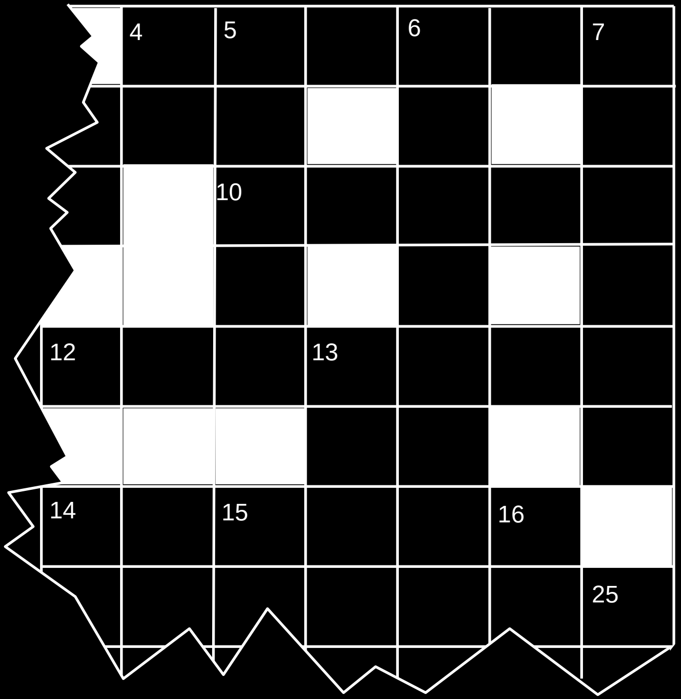
E. I’m not sure



Decoding by belief propagation

Crossword puzzles

- Iterate!



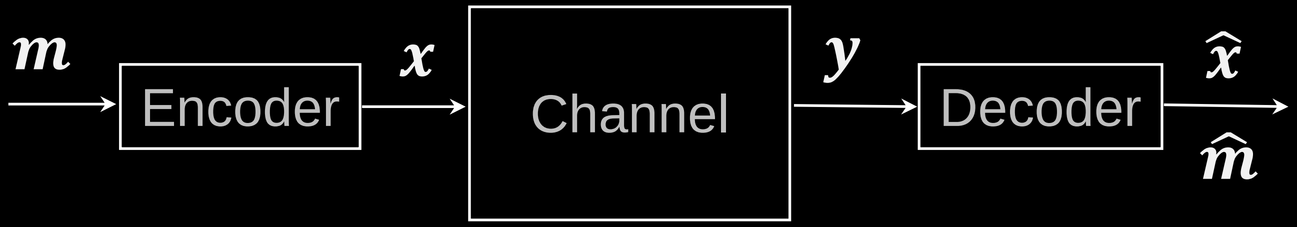
Across:

4 Animal with long ears and a short tail.
10 Person who is in charge of a country.
12 In no place.

Down:

5 Pointer, weapon fired from a bow.
6 Accept as true.
7 A place to shoot at; objective.

Maximum likelihood (ML) decoding



- ML decoding rule:

$$\hat{x} = \underset{x \in C}{\operatorname{argmax}}\ P(x|y)$$

- Must evaluate posterior for each of the 2^k codewords!
- Make bit-wise decisions instead:

$$\hat{x}_j = \underset{x_j \in \{0,1\}}{\operatorname{argmax}}\ P(x_j|y) = \underset{x_j \in \{0,1\}}{\operatorname{argmax}} \sum_{x_1,\ldots,x_{j-1},x_{j+1},\ldots,x_n \in \{0,1\}^{n-1}} P(x|y)$$

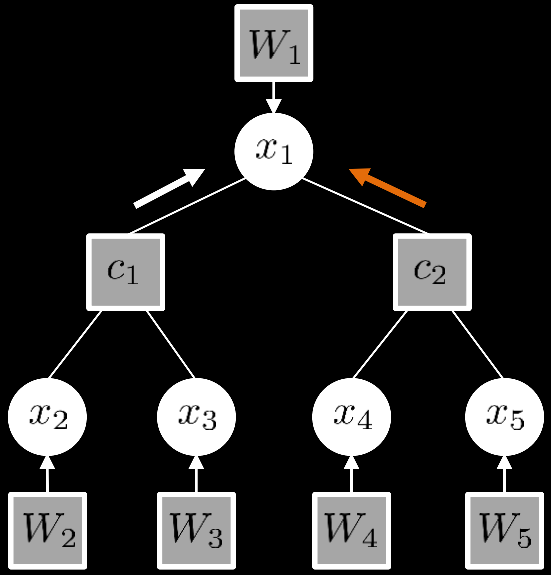
Bit-wise maximum likelihood (ML) decoding

$$\hat{x}_1 = \underset{x_1\in\{0,1\}}{\text{argmax}}\ P(x_1|y) = \underset{x_1\in\{0,1\}}{\text{argmax}} \sum_{x_2,x_3,x_4,x_5\in\{0,1\}^4} P(x|y)$$

$$= \underset{x_1\in\{0,1\}}{\text{argmax}} \sum_{x_2,x_3,x_4,x_5\in\{0,1\}^4} P(y|x)P(x)$$

$$= \underset{x_1\in\{0,1\}}{\text{argmax}} \sum_{x_2,x_3,x_4,x_5\in\{0,1\}^4} \left(\prod_{j=1}^{5} W_j\right) \mathbb{I}(c_1=0)\mathbb{I}(c_2=0)$$

$$= \underset{x_1\in\{0,1\}}{\text{argmax}}\ W_1 \cdot \boxed{\left(\sum_{x_2,x_3\in\{0,1\}^2} \mathbb{I}(c_1=0)\cdot W_2\cdot W_3\right)}\ \boxed{\left(\sum_{x_4,x_5\in\{0,1\}^2} \mathbb{I}(c_2=0)\cdot W_4\cdot W_5\right)}$$



$$W_j = P(y_j|x_j)$$

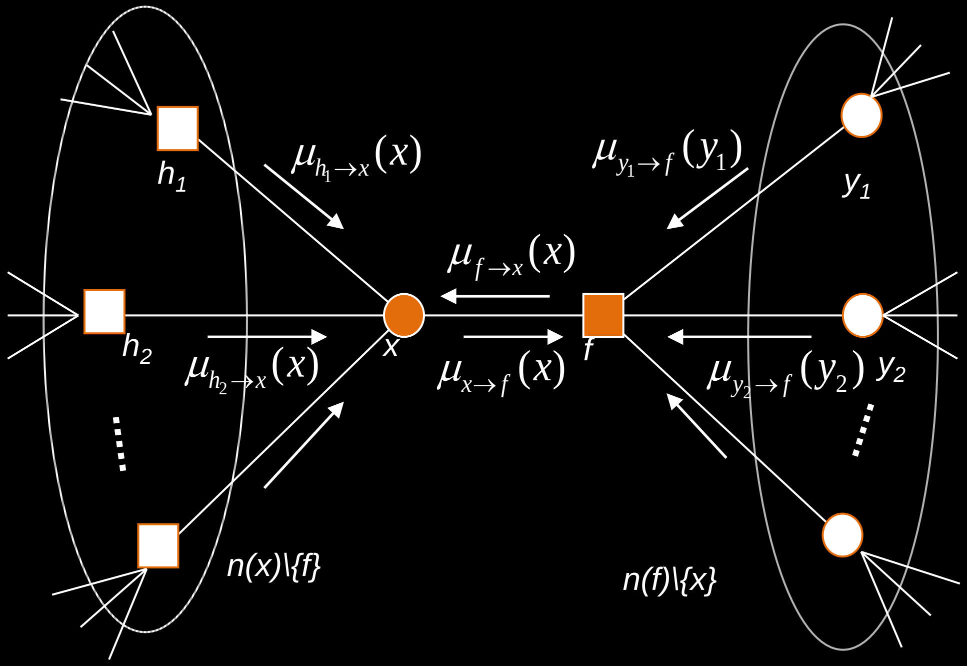
$$c_1 :\ x_1 + x_2 + x_3 = 0$$

$$c_2 :\ x_1 + x_4 + x_5 = 0$$

Distributivity of addition over multiplication!

Sum-product algorithm (SPA)

Belief propagation (BP)



Variable node (VN) update:

$$\mu_{x\to f}(x) = \prod_{h\in n(x)\backslash\{f\}} \mu_{h\to x}(x)$$

Check node (CN) update:

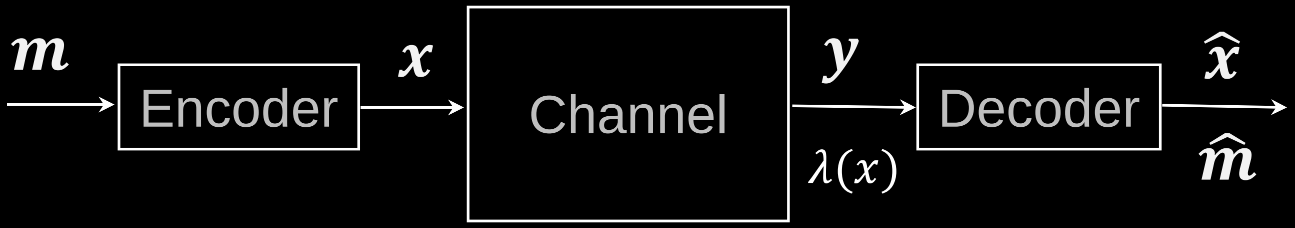
$$\mu_{f\to x}(x) = \sum_{\sim\{x\}} \left(f(X) \prod_{h\in n(f)\backslash\{x\}} \mu_{y\to f}(y) \right)$$

Variable node (VN) decision:

$$g_i(x_i) = \prod_{h\in n(x_i)} \mu_{h\to x_i}(x_i)$$

Decoders for channels with soft outputs

- In addition to the channel value, a measure of bit reliability is also provided


$\boldsymbol{m}$

Encoder

$\boldsymbol{x}$

Channel

$\boldsymbol{y}$

$\lambda(x)$

Decoder

$\hat{\boldsymbol{x}}$

$\hat{\boldsymbol{m}}$

- Bit log-likelihood ratio (LLR) given y_i:

$$\lambda(x_i) = \log\frac{P(x_i = 0|y_i)}{P(x_i = 1|y_i)}$$

$$= \log\frac{\frac{p(y_i|x_i = 0)P(x_i = 0)}{p(y_i)}}{\frac{p(y_i|x_i = 1)P(x_i = 1)}{p(y_i)}} \quad = \log\frac{p(y_i|x_i = 0)P(x_i = 0)}{p(y_i|x_i = 1)P(x_i = 1)}$$

$$= \log\frac{p(y_i|x_i = 0)}{p(y_i|x_i = 1)} + \log\frac{P(x_i = 0)}{P(x_i = 1)}$$

Log-likelihood ratio (LLR)

- Without prior knowledge on x_i:

$$\gamma_i = \lambda(x_i) = \log \frac{p(y_i|x_i = 0)}{p(y_i|x_i = 1)}$$

- For AWGN ($y_i = x_i + n_i; \quad n_i \sim N(0,1)$):

$$\gamma_i = \log \frac{p(y_i|x_i = 0)}{p(y_i|x_i = 1)} = \frac{1}{2\sigma^2}\left(-(y_i - 1)^2 + (y_i + 1)^2\right) = \frac{y_i}{2\sigma^2}$$

- For BSC with parameter α:

$$\gamma_i = \begin{cases} \log \dfrac{1-\alpha}{\alpha} & \text{if } y_i = 0 \\ \log \dfrac{1-\alpha}{\alpha} & \text{if } y_i = 1 \end{cases}$$

BP or SPA

- The update rule

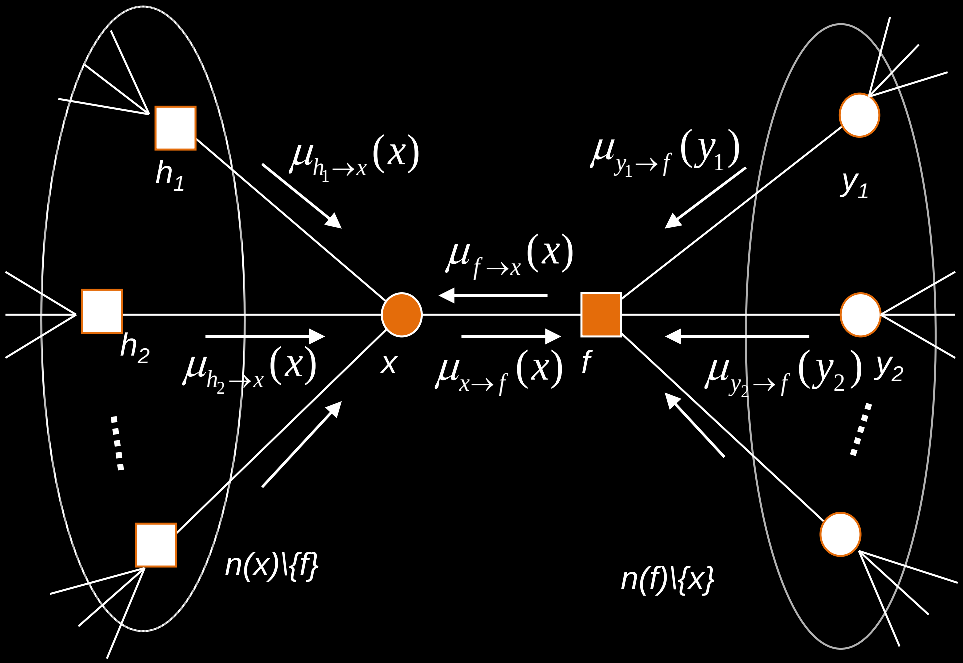
$$\begin{aligned}\omega_{i\to\alpha}^{(0)} &= \gamma_i \\ \varpi_{\alpha\to i}^{(k)} &= 2\tanh^{-1}\left(\prod_{j\in\mathcal{N}(\alpha)\backslash i}\tanh\left(\frac{1}{2}\omega_{j\to\alpha}^{(k-1)}\right)\right) \\ \omega_{i\to\alpha}^{(k)} &= \gamma_i + \sum_{\delta\in\mathcal{N}(i)\backslash\alpha}\varpi_{\delta\to i}^{(k)}\end{aligned}$$

- The result of decoding after k iterations, denoted by $\mathbf{x}^{(k)}$ is determined by the sign of

$$m_i^{(k)} = \gamma_i + \textstyle\sum_{\alpha\in\mathcal{N}(i)}\varpi_{\alpha\to i}^{(k)}$$

$$\text{If } m_i^{(k)} > 0 \text{ then } x_i^{(k)} = 0, \text{ otherwise } x_i^{(k)} = 1$$

The min-sum approximation (MSA)



$$\mu_{x \to f} = \lambda_x + \sum_{h \in n(x) \backslash \{f\}} \mu_{h \to f}$$

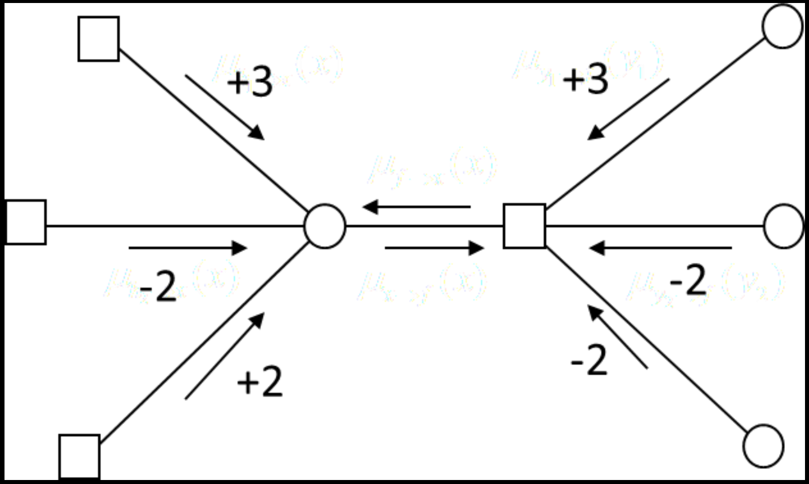
$$\mu_{f \to x}(x) = \prod_{y \in n(f) \backslash \{x\}} \operatorname{sgn}(\mu_{y \to f}) \min_{y \in n(f) \backslash \{x\}} |\mu_{y \to f}|$$

$$g_i(x_i) = \lambda(x_i) + \sum_{h \in n(x_i)} \mu_{h \to x_i}$$

Poll Question 11

The min-sum decoder operates by sending messages between variable and check node processing units. After receiving all three messages from its neighboring checks, assuming that the channel value is -3, the variable node processing unit of the variable shown in the picture below will send the following message to the check node processing unit of the remaining check:

A. $\text{sign}(+3)\text{sign}(-2)\text{sign}(-2)\min(|+3|,|-2|,|-2| = +2$
B. $\text{sign}(+3)\text{sign}(-2)\text{sign}(+2)\min(|+3|,|-2|,|+2| = -2$
C. $-1 + \text{sign}(+3)\text{sign}(-2)\text{sign}(-2)\min(|+3|,|-2|,|-2| = -3 + 2 = -1$
D. $-1 + \text{sign}(+3)\text{sign}(-2)\text{sign}(+2)\min(|+3|,|-2|,|+2| = -3 - 2 = -5$
E. ∞
F. -∞
G. 0
H. -4
I. +4
J. I’m not sure

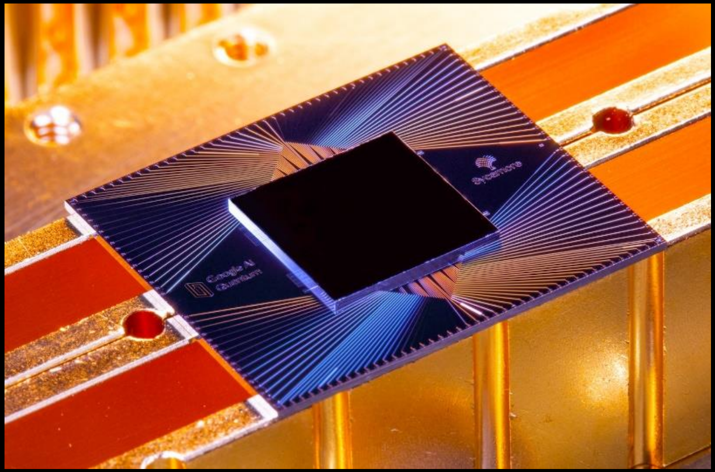


Applications of LDPC codes

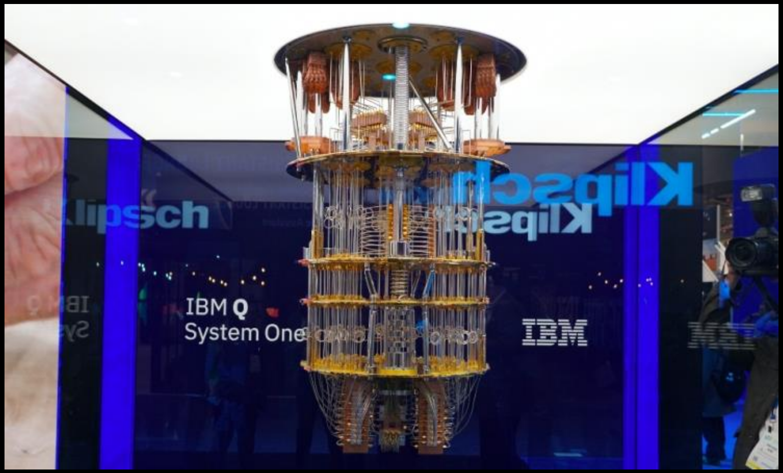
- Wireless networks, satellite communications, deep-space communications, power line communications
- Magnetic hard disk drives, optical communications, flash memories
- Standards include:
  - Digital video broadcast over satellite (DVB-S2 Standard) and over cable (DVB-C2 Standard), terrestrial television broadcasting (DVB-T2, DVB-T2-Lite Standards)
  - GEO-Mobile Radio (GMR) satellite telephony (GMR-1 Standard), local and metropolitan area networks (LAN/MAN) (IEEE 802.11 (WiFi))
  - Wireless personal area networks (WPAN) (IEEE 802.15.3c (60 GHz PHY)), wireless local and metropolitan area networks (WLAN/WMAN) (IEEE 802.16 (Mobile WiMAX)
  - Near-earth and deep space communications (CCSDS), wire and power line communications (ITU-T G.hn (G.9960))
  - Ultra-wide band technologies (WiMedia 1.5 UWB)

Quantum Fundamentals

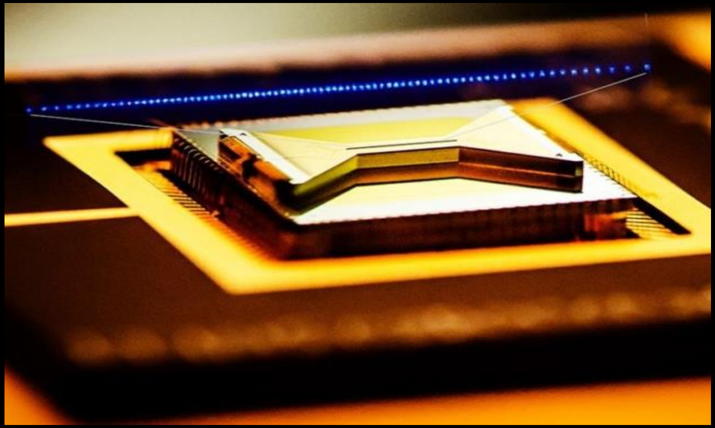
Quantum Technologies



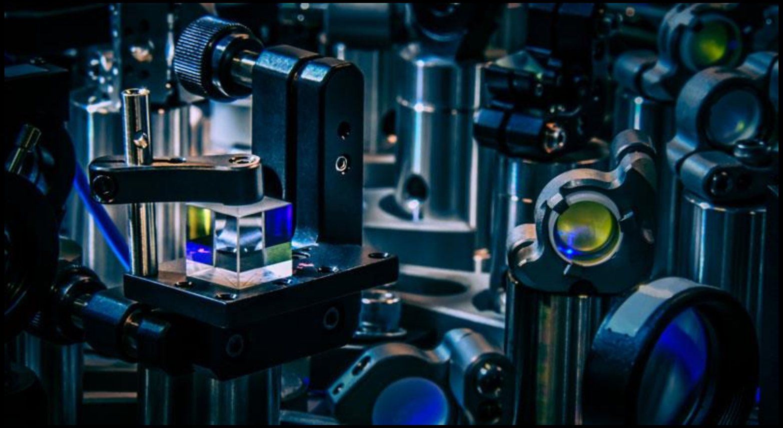
Google



IBM



IonQ



Quantinuum

A qubit is a 2-dimensional vector

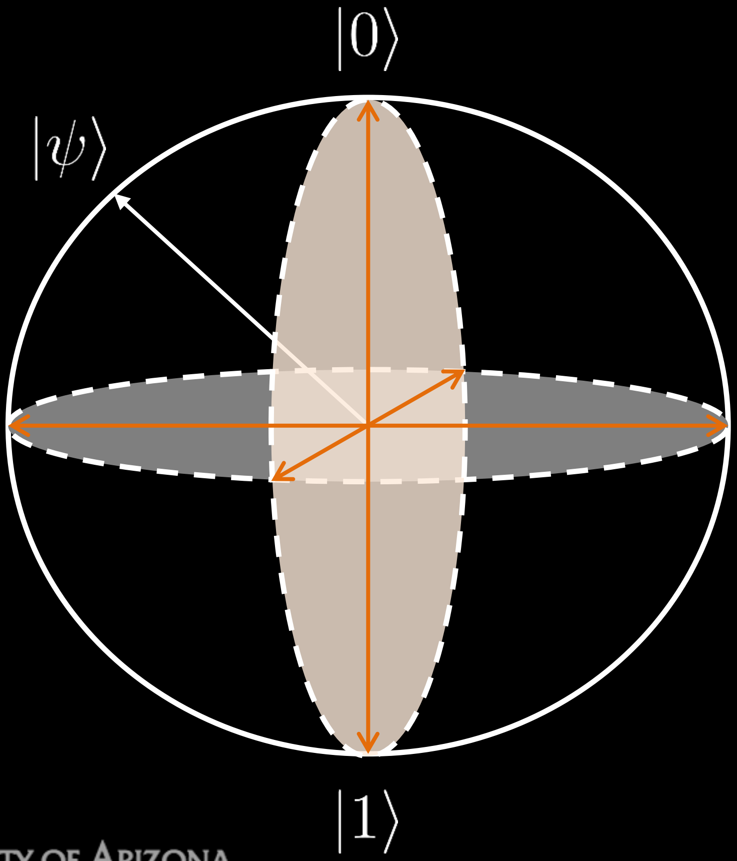
- Computational basis states: “Ket 0” , “Ket 1”

Dirac notation: $|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix},\ |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$

- A single-qubit state:

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$$

$$\alpha, \beta \in \mathbb{C},\ |\alpha|^2 + |\beta|^2 = 1$$



Bloch Sphere
Visualizing 1 qubit

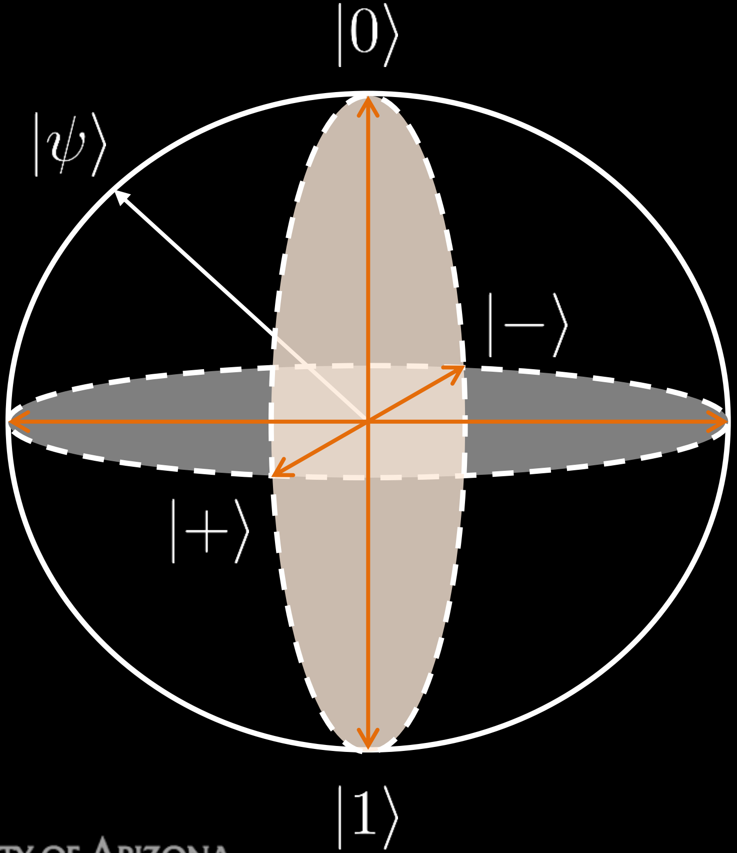
Rotating to the conjugate basis

- Conjugate basis states: "Ket +" , "Ket –"

Dirac notation: $$|+\rangle = \frac{1}{\sqrt{2}}\begin{bmatrix}1\\1\end{bmatrix},\ |-\rangle = \frac{1}{\sqrt{2}}\begin{bmatrix}1\\-1\end{bmatrix}$$

- A single-qubit state:

$$\begin{aligned}|\psi\rangle &= \alpha\,|0\rangle + \beta\,|1\rangle \\ &= \gamma\,|+\rangle + \delta\,|-\rangle\end{aligned}$$



Bloch Sphere
Visualizing 1 qubit

What can you do with a qubit?

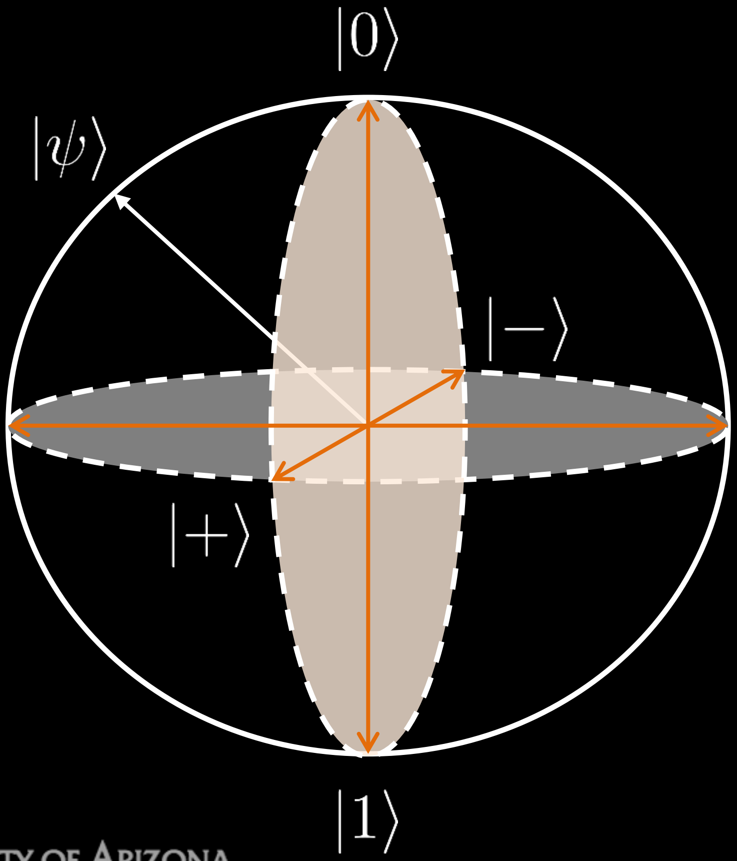
- Unitary operations: complex rotations, reversible

$$U \in \mathbb{U}^{2\times 2}: U^{-1} = U^\dagger$$ Hermitian transpose

- Measurement:
  project the state on a basis, irreversible

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$$
$$= \gamma\,|+\rangle + \delta\,|-\rangle$$

Bloch Sphere
Visualizing 1 qubit



Single Qubit: Unitary Operations

First quantum operation – Hadamard "gate"

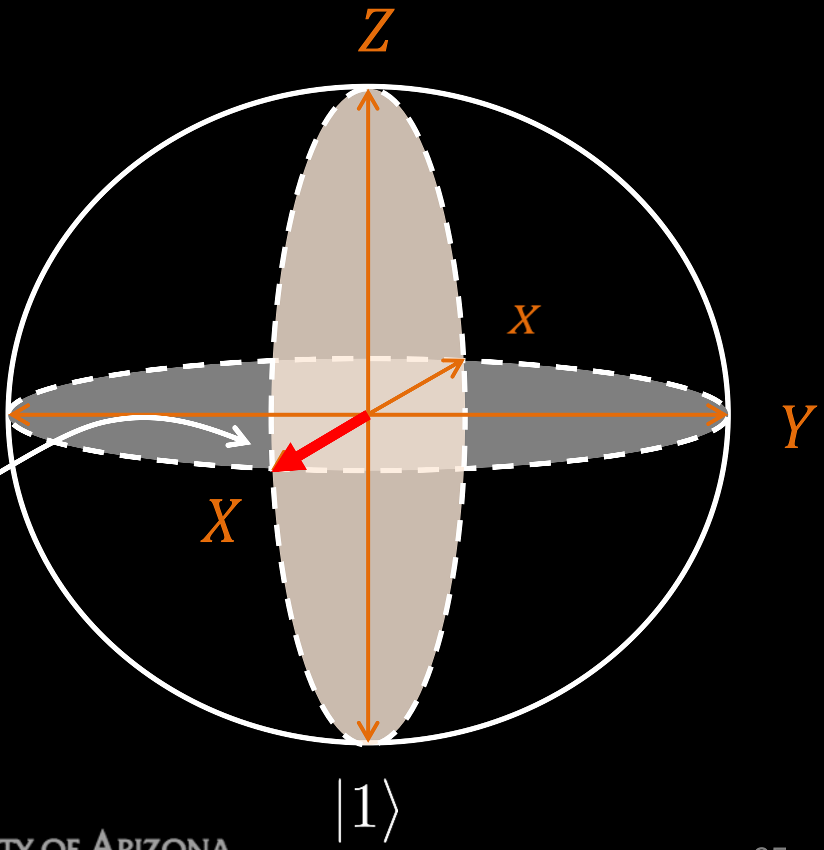
- Switches between computational and conjugate bases

$$H\left|0\right\rangle = \left|+\right\rangle\ ,\ \ H\left|1\right\rangle = \left|-\right\rangle$$
$$H\left|+\right\rangle = \left|0\right\rangle\ ,\ \ H\left|-\right\rangle = \left|1\right\rangle$$

- Matrix representation:

$$H = \frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}$$
$$H^{-1} = H^{\dagger} = H$$

Take any initial state



First quantum operation – Hadamard “gate”

- Switches between computational and conjugate bases

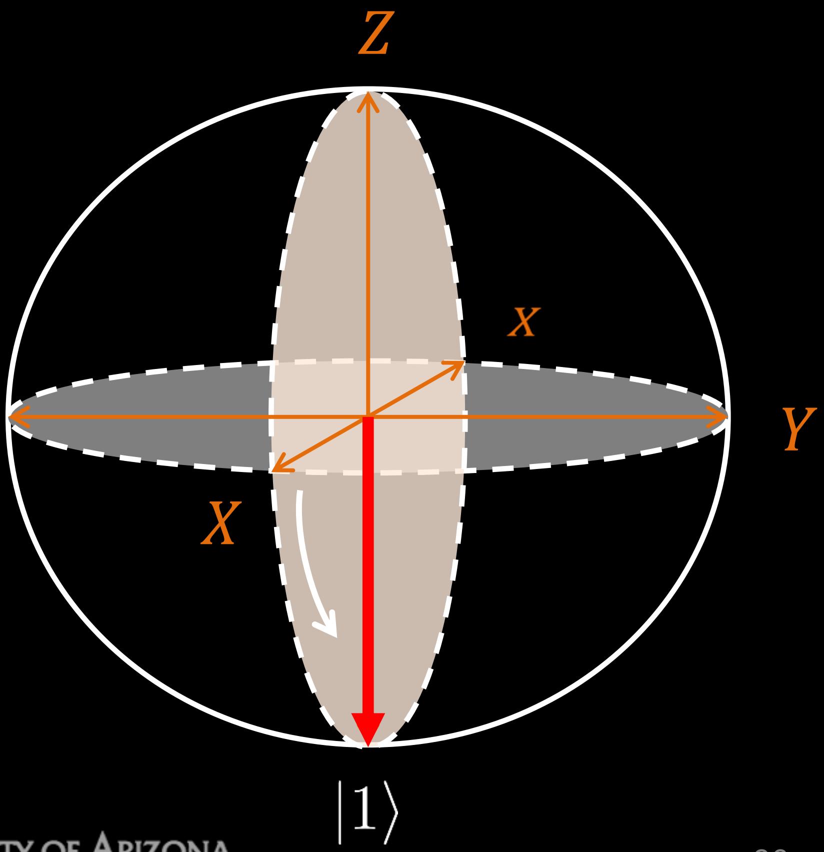
$$H\,|0\rangle = |+\rangle\ ,\ H\,|1\rangle = |-\rangle$$
$$H\,|+\rangle = |0\rangle\ ,\ H\,|-\rangle = |1\rangle$$

- Matrix representation:

$$H = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$
$$H^{-1} = H^{\dagger} = H$$

Take any initial state

Rotate 90° by Y axis



First quantum operation – Hadamard "gate"

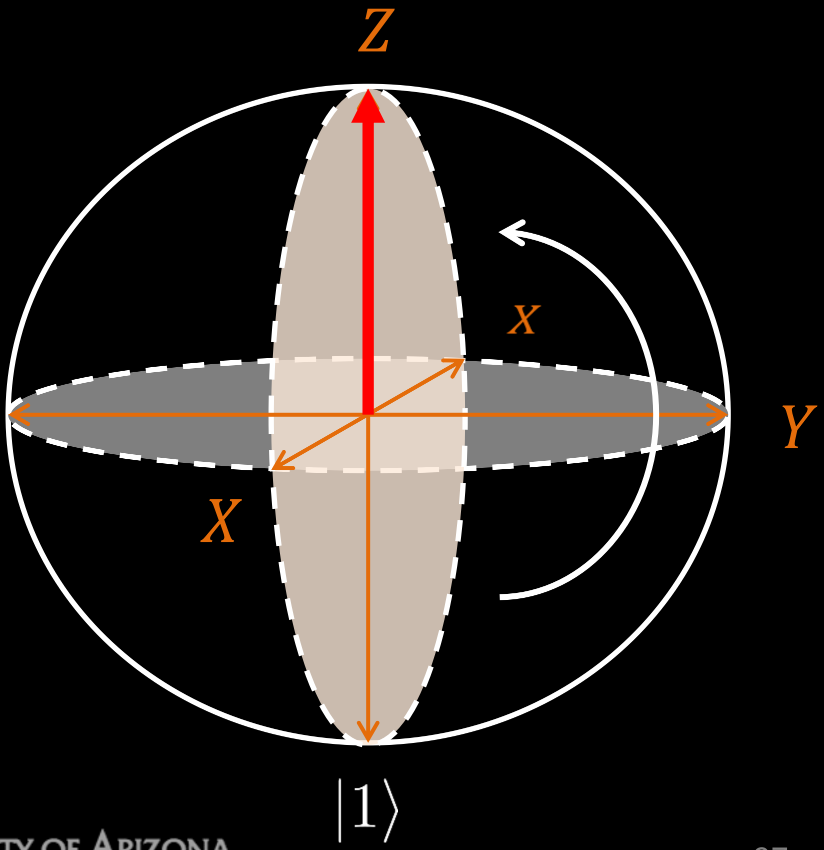
- Switches between computational and conjugate bases

$$H\,|0\rangle = |+\rangle\ ,\ \ H\,|1\rangle = |-\rangle$$
$$H\,|+\rangle = |0\rangle\ ,\ \ H\,|-\rangle = |1\rangle$$

- Matrix representation:

$$H = \frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}$$
$$H^{-1} = H^{\dagger} = H$$

Take any initial state
Rotate 90° by Y axis
Then rotate 180° by X axis



Poll Question 12

If we start with the single qubit state $|0\rangle$ and apply the H gate twice, what is the resulting state?

A. $|+\rangle$
B. $|-\rangle$
C. $|0\rangle$
D. $|1\rangle$
E. None of the above

F. I’m not sure

Pauli rotations

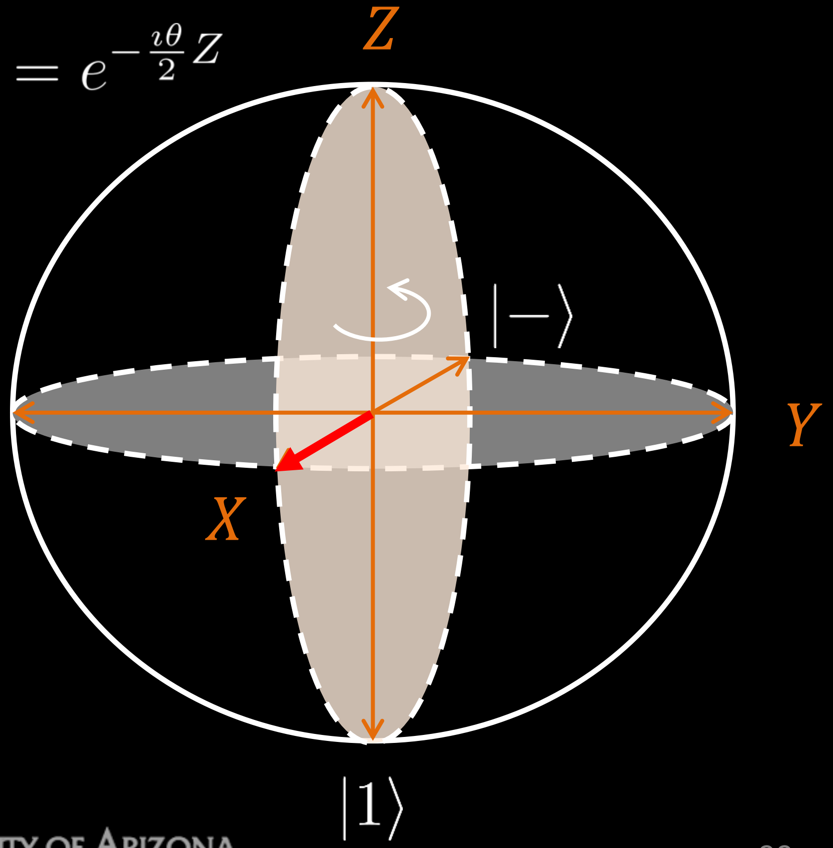
- The single-qubit Pauli matrices are: $(\imath = \sqrt{-1})$

$$I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, Y = \begin{bmatrix} 0 & -\imath \\ \imath & 0 \end{bmatrix}$$

- These are π-rotations: $R_Z(\theta) = e^{-\frac{\imath\theta}{2}Z}$

$$e^{-\frac{\imath\pi}{2}Z} = \begin{bmatrix} e^{-\frac{\imath\pi}{2}} & 0 \\ 0 & e^{\frac{\imath\pi}{2}} \end{bmatrix}$$
$$= e^{-\frac{\imath\pi}{2}} \begin{bmatrix} 1 & 0 \\ 0 & e^{\imath\pi} \end{bmatrix}$$
$$\equiv Z$$

Global phases don’t matter!



Pauli rotations

- The single-qubit Pauli matrices are: $(\imath = \sqrt{-1})$

$$I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, Y = \begin{bmatrix} 0 & -\imath \\ \imath & 0 \end{bmatrix}$$

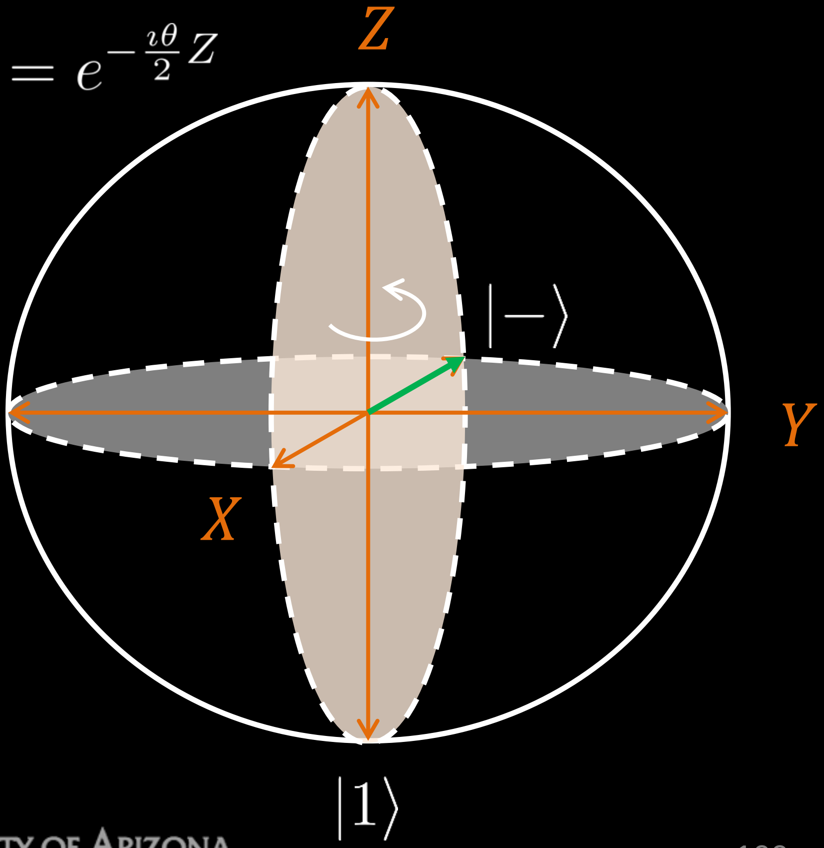
- These are π-rotations: $R_Z(\theta) = e^{-\frac{\imath\theta}{2}Z}$

$$e^{-\frac{\imath\pi}{2}Z} = \begin{bmatrix} e^{-\frac{\imath\pi}{2}} & 0 \\ 0 & e^{\frac{\imath\pi}{2}} \end{bmatrix}$$

$$= e^{-\frac{\imath\pi}{2}} \begin{bmatrix} 1 & 0 \\ 0 & e^{\imath\pi} \end{bmatrix}$$

$$\equiv Z$$

$$Z\,|+\rangle = |-\rangle \;,\; Z\,|-\rangle = |+\rangle$$



Pauli rotations

- The single-qubit Pauli matrices are: $(\imath = \sqrt{-1})$

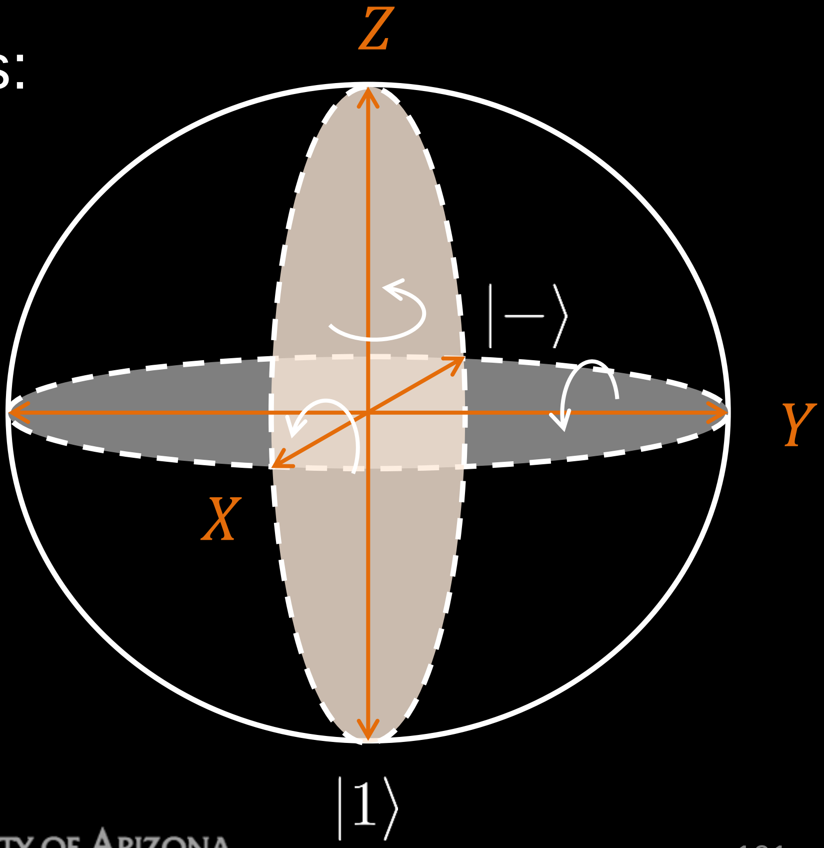
$$I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, Y = \begin{bmatrix} 0 & -\imath \\ \imath & 0 \end{bmatrix}$$

- Bit- and Phase-flip operations:

$$X\left|0\right\rangle = X\begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \left|1\right\rangle$$

$$Z\left|0\right\rangle = \left|0\right\rangle \ , \ Z\left|1\right\rangle = -\left|1\right\rangle$$

$Y = \imath XZ$ (Bit-Phase flip)



Pauli rotations

- The single-qubit Pauli matrices are: $(\imath = \sqrt{-1})$

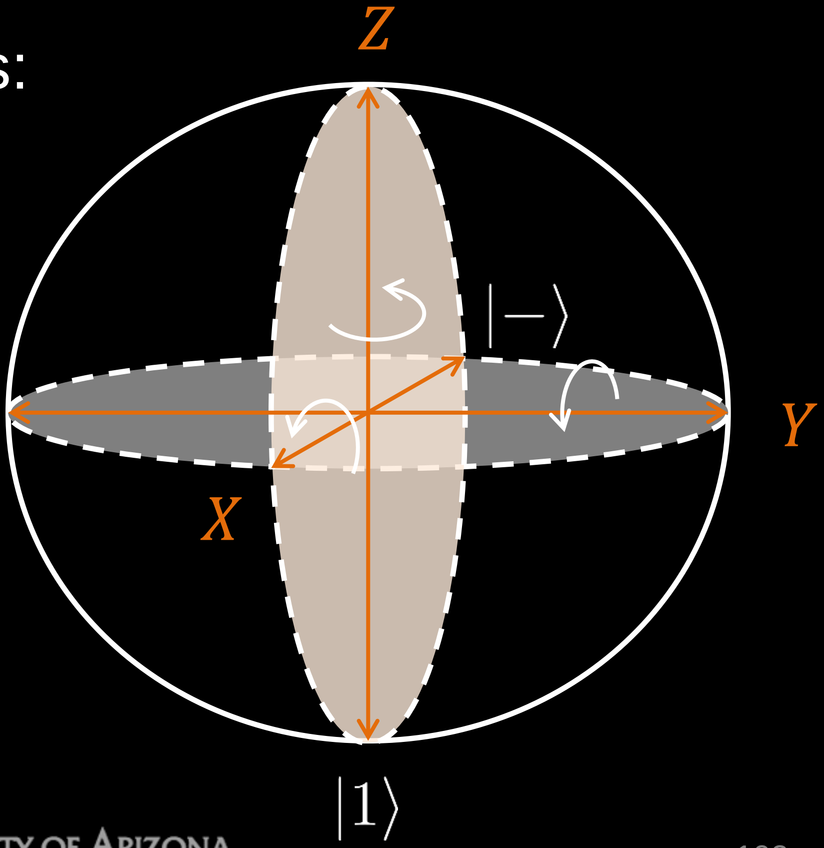
$$I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, Y = \begin{bmatrix} 0 & -\imath \\ \imath & 0 \end{bmatrix}$$

- Bit- and Phase-flip operations:

$$Z\left|+\right\rangle \propto Z \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ -1 \end{bmatrix} = \left|-\right\rangle$$

$$X\left|+\right\rangle = \left|+\right\rangle, \; X\left|-\right\rangle = -\left|-\right\rangle$$

$Y = \imath XZ$ (Bit-Phase flip)



Arbitrary single-qubit gate

- How can we implement an arbitrary unitary operation?
- Classical Computing: NAND and NOR are universal
- Quantum Computing: A finite but universal gate set?

Universal gate set for one qubit

- The single-qubit Pauli gates are: $(\imath = \sqrt{-1})$

$$I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, Y = \begin{bmatrix} 0 & -\imath \\ \imath & 0 \end{bmatrix}$$

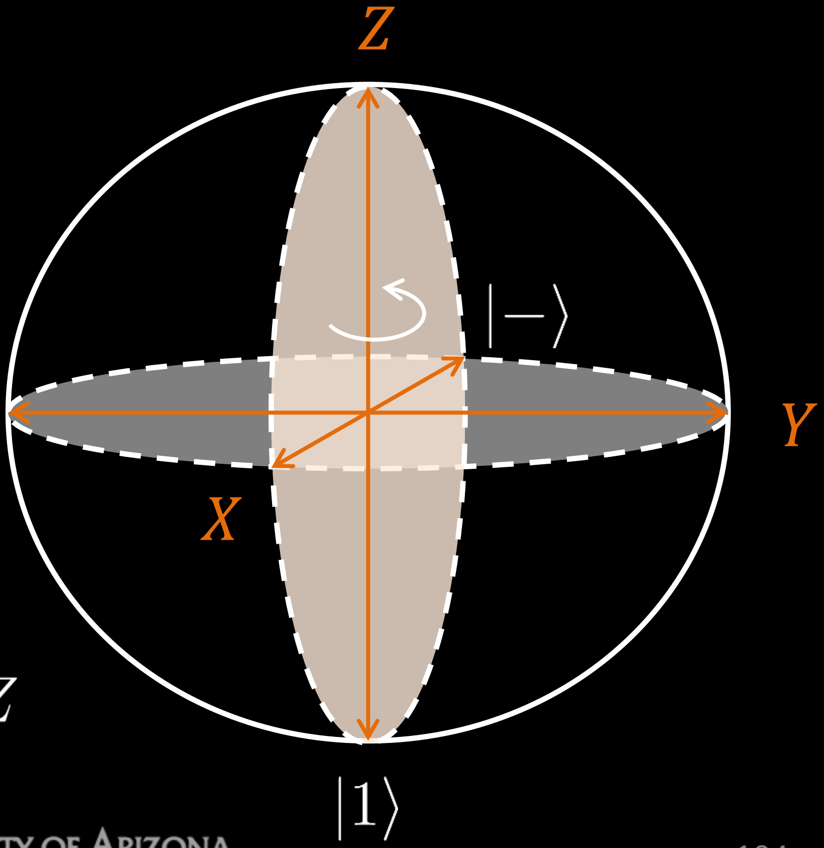
- Hadamard gate:

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} = \frac{X+Z}{\sqrt{2}}$$

- T gate ($\pi/4$-rotation):

$$T = \begin{bmatrix} 1 & 0 \\ 0 & e^{\imath\pi/4} \end{bmatrix} \equiv e^{-\frac{\imath\pi}{8}Z}$$

$$T^4 = Z\,,\; HZH = X\,,\; Y = \imath XZ$$



Poll Question 13

The Phase gate is defined by $P = \sqrt{Z} = T^2$. What is the matrix representation of the Phase gate?

$A.\quad P = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$

$B.\quad P = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$

$C.\quad P = \begin{bmatrix} 1 & 0 \\ 0 & -\imath \end{bmatrix}$

$D.\quad P = \begin{bmatrix} 1 & 0 \\ 0 & \imath \end{bmatrix}$

E. None of the above

F. I’m not sure

Single Qubit: Measurements

Projective measurement

- Measure Z on $|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$; "Bra psi" $\langle\psi| = |\psi\rangle^\dagger$

1. First, diagonalize the measured "observable"

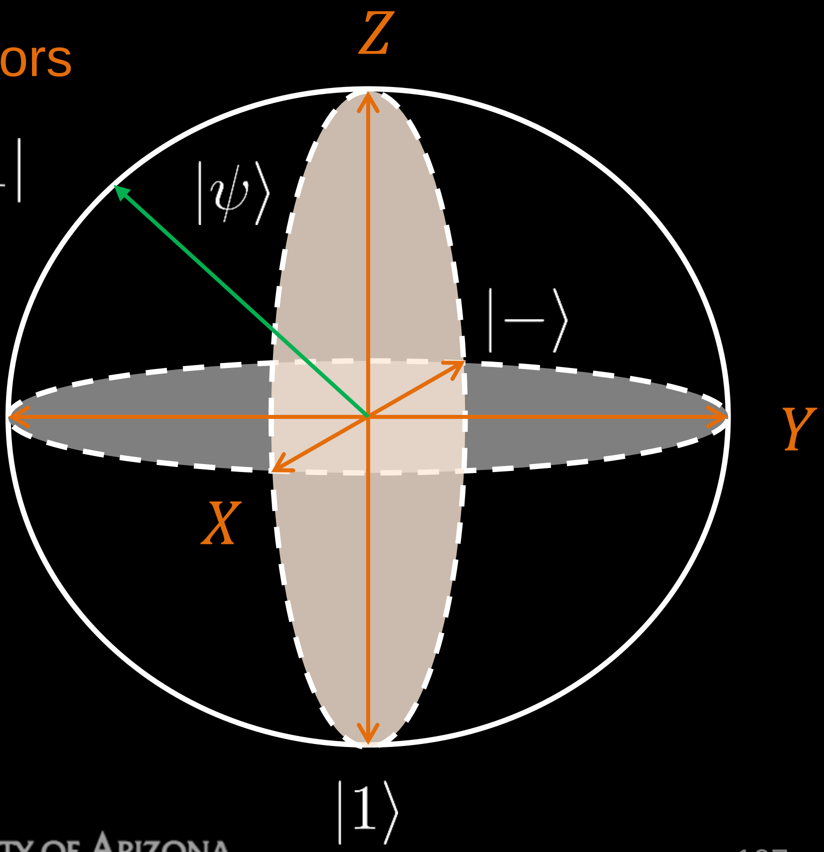
$$Z = |0\rangle\langle 0| - |1\rangle\langle 1|$$

2. Define projectors from eigenvectors

$$M_{+1} = |0\rangle\langle 0| \;,\; M_{-1} = |1\rangle\langle 1|$$

3. Possible outcomes "+1", "−1"

$$\mathbb{P}[+1] = \langle\psi|\,M_{+1}\,|\psi\rangle = |\alpha|^2$$
$$\mathbb{P}[-1] = \langle\psi|\,M_{-1}\,|\psi\rangle = |\beta|^2$$



Projective measurement

- Measure Z on $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$; “Bra psi” $\langle\psi| = |\psi\rangle^\dagger$

1. First, diagonalize the measured “observable”

$$Z = |0\rangle\langle 0| - |1\rangle\langle 1|$$

2. Define projectors from eigenvectors

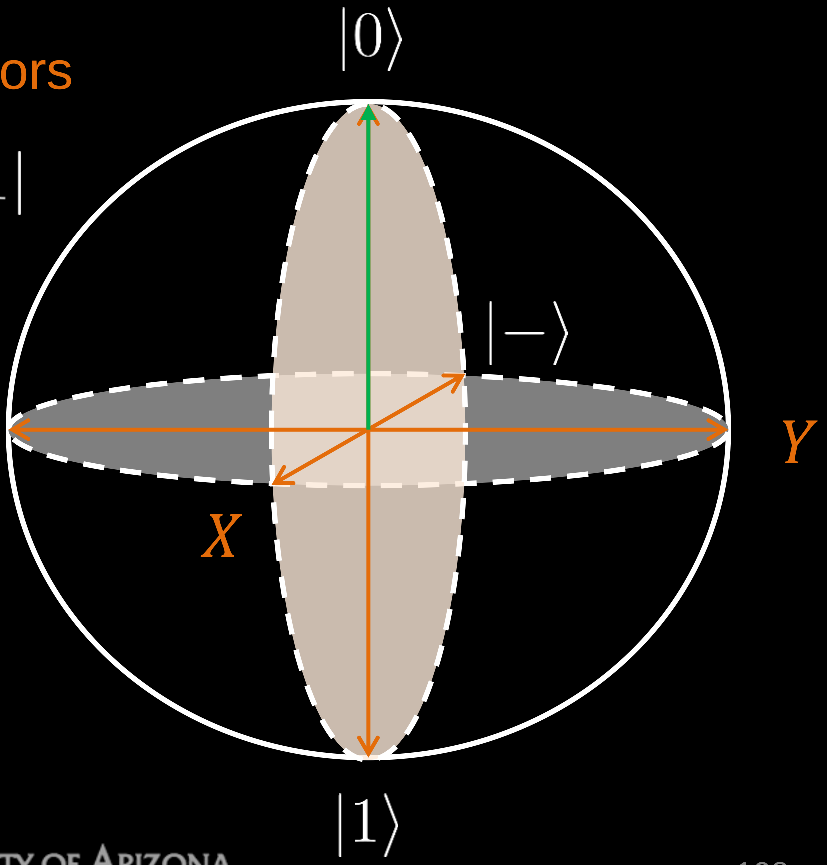
$$M_{+1} = |0\rangle\langle 0| \;,\; M_{-1} = |1\rangle\langle 1|$$

3. Possible outcomes “$+1$”, “-1”

$$\mathbb{P}[+1] = \langle\psi| M_{+1} |\psi\rangle = |\alpha|^2$$
$$\mathbb{P}[-1] = \langle\psi| M_{-1} |\psi\rangle = |\beta|^2$$

4. Post-measurement state

$$|\psi_\pm\rangle = \frac{M_{\pm 1}|\psi\rangle}{\sqrt{\mathbb{P}[\pm 1]}} = |0/1\rangle$$



Information storage in a qubit

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$$

$$\alpha, \beta \in \mathbb{C},\ \ |\alpha|^2 + |\beta|^2 = 1$$

Technically, a qubit can store infinite information!

But NO measurement can retrieve it exactly!

This is true INDEPENDENT of the measurement basis

Projective measurement

- Measure X on $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle = \gamma|+\rangle + \delta|-\rangle$

1. First, diagonalize the measured "observable"

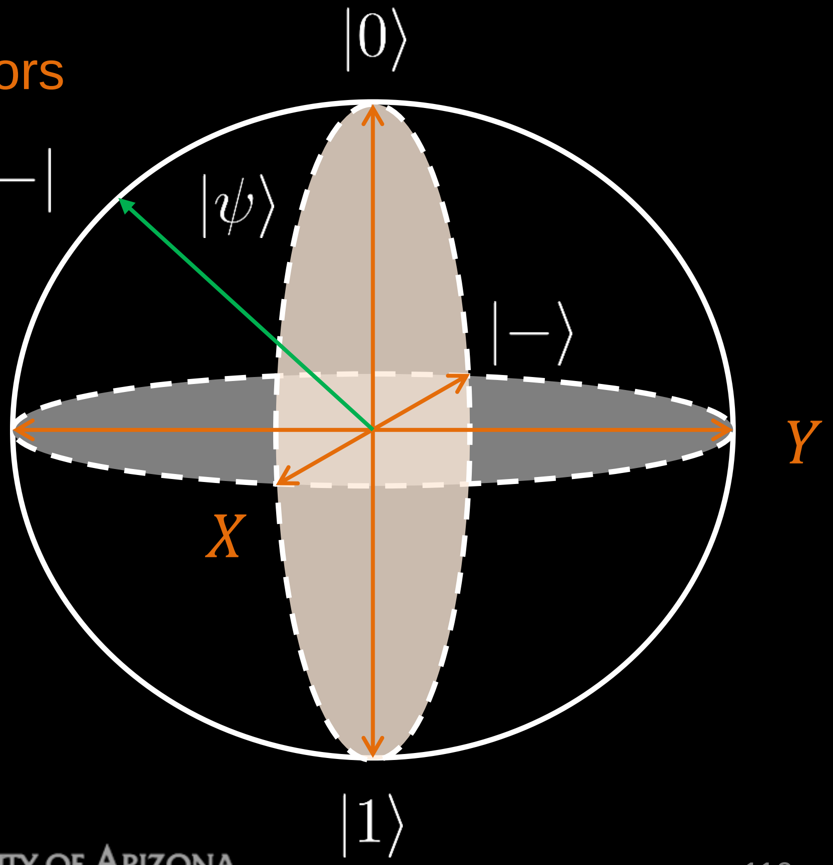
$$X = |+\rangle\langle+| - |-\rangle\langle-|$$

2. Define projectors from eigenvectors

$$M_{+1} = |+\rangle\langle+| \, , \, M_{-1} = |-\rangle\langle-|$$

3. Possible outcomes "+1", "−1"

$$\mathbb{P}[+1] = \langle\psi| M_{+1} |\psi\rangle = |\gamma|^2$$
$$\mathbb{P}[-1] = \langle\psi| M_{-1} |\psi\rangle = |\delta|^2$$



Projective measurement

- Measure X on $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle = \gamma|+\rangle + \delta|-\rangle$

1. First, diagonalize the measured "observable"

$$X = |+\rangle\langle+| - |-\rangle\langle-|$$

2. Define projectors from eigenvectors

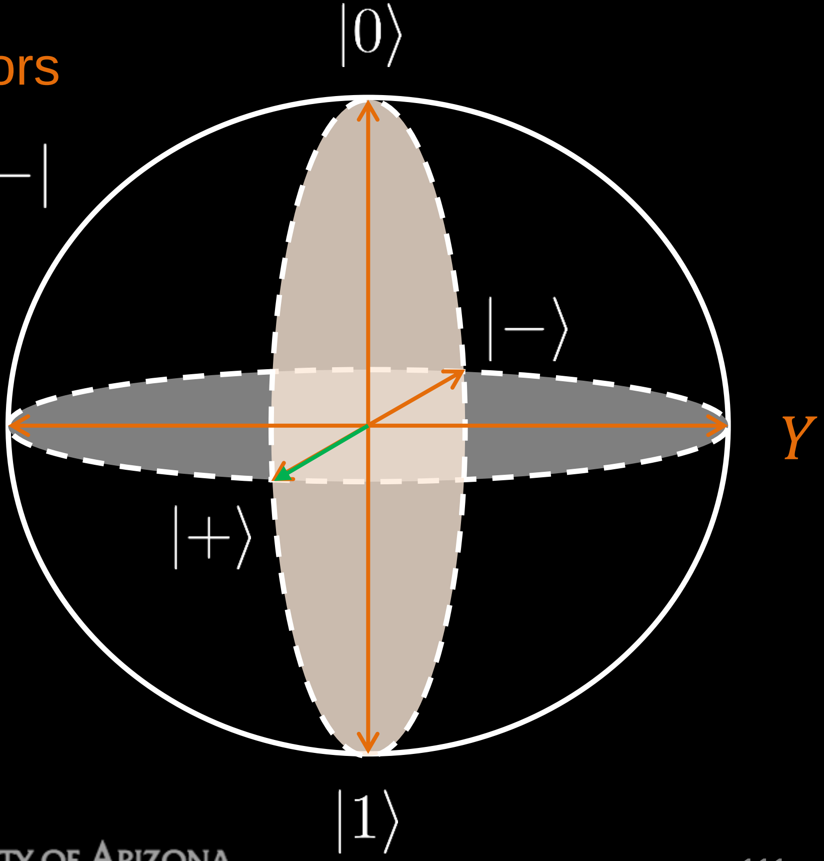
$$M_{+1} = |+\rangle\langle+| \,,\; M_{-1} = |-\rangle\langle-|$$

3. Possible outcomes "+1", "−1"

$$\mathbb{P}[+1] = \langle\psi| M_{+1} |\psi\rangle = |\gamma|^2$$
$$\mathbb{P}[-1] = \langle\psi| M_{-1} |\psi\rangle = |\delta|^2$$

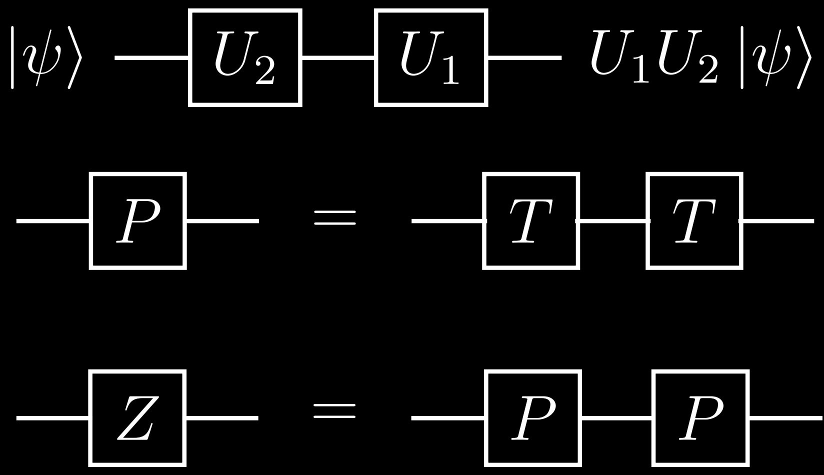
4. Post-measurement state

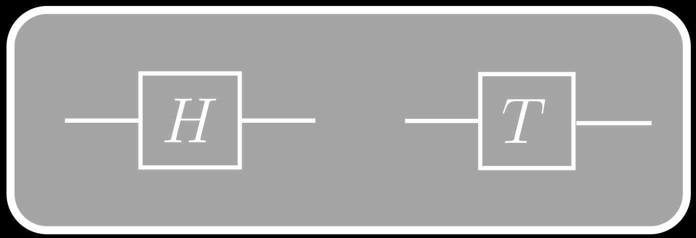
$$|\psi_\pm\rangle = \frac{M_{\pm1}|\psi\rangle}{\sqrt{\mathbb{P}[\pm1]}} = |\pm\rangle$$



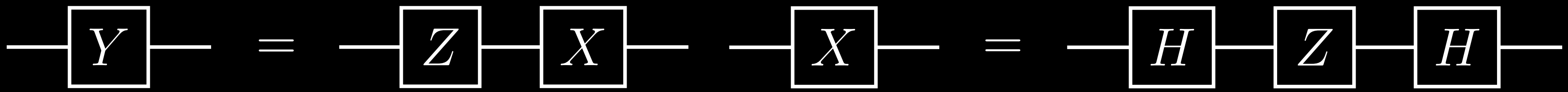
Quantum circuit notation

- Single-qubit gates

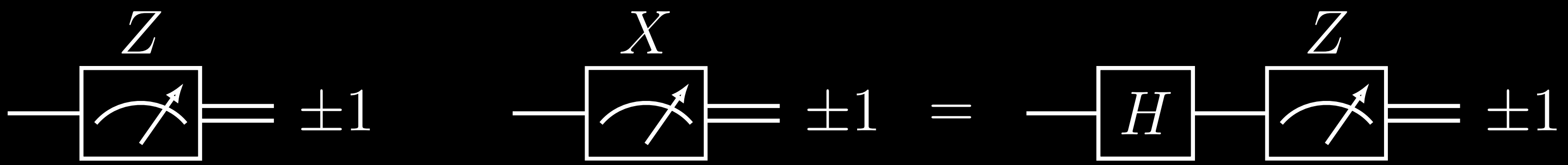




Universal Set



- Single-qubit measurements



Poll Question 14

Let the initial state be $|0\rangle$. We apply the H gate and then measure in the Z basis. What is the probability of the measurement result -1 and what is the corresponding post-measurement state?

A. $\frac{1}{2}$ and $|1\rangle$

B. $\frac{1}{\sqrt{2}}$ and $|1\rangle$

C. $\frac{1}{2}$ and $|0\rangle$

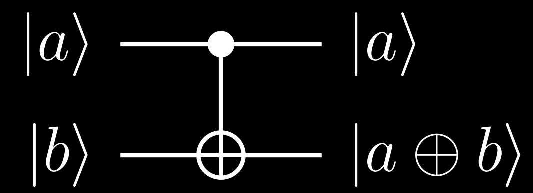
D. $\frac{1}{\sqrt{2}}$ and $|0\rangle$

E. I’m not sure

Multiple Qubits

Moving beyond one qubit

Controlled-NOT gate: Flip target qubit if control qubit is 1



Control qubit $|a\rangle$ → $|a\rangle$

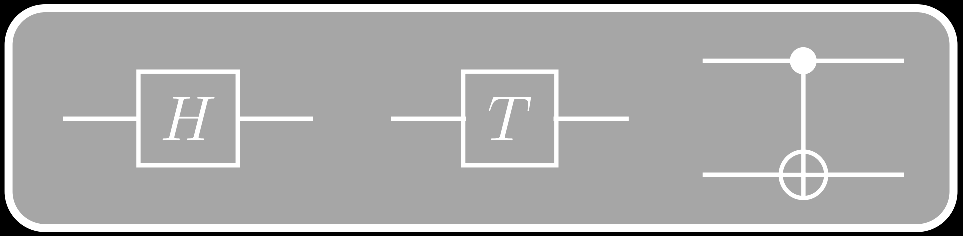
Target qubit $|b\rangle$ → $|a \oplus b\rangle$

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

CX

Input (a, b)	Output $(a, a \oplus b)$
00	00
01	01
10	11
11	10

Universal gates on n qubits:



Kronecker (or Tensor) product

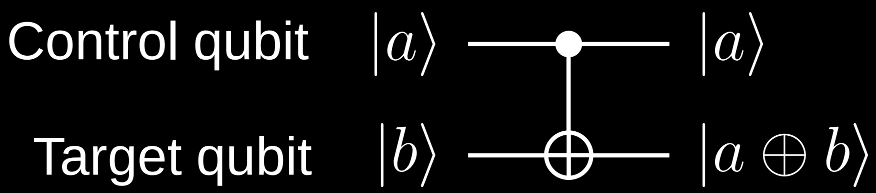
$$A_{m\times n}\otimes B_{p\times q}=\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}\otimes\begin{bmatrix} b_{11} & b_{12} & \cdots & b_{1q} \\ b_{21} & b_{22} & \cdots & b_{2q} \\ \vdots & \vdots & \ddots & \vdots \\ b_{p1} & b_{p2} & \cdots & b_{pq} \end{bmatrix}$$

$$=\begin{bmatrix} a_{11}\times\begin{bmatrix} b_{11} & \cdots & b_{1q} \\ \vdots & \ddots & \vdots \\ b_{p1} & \cdots & b_{pq} \end{bmatrix} & a_{12}\times B & \cdots & a_{1n}\times B \\ a_{21}\times B & a_{22}\times B & \cdots & a_{2n}\times B \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1}\times B & a_{m2}\times B & \cdots & a_{mn}\times B \end{bmatrix}_{mp\times nq}$$

Useful Property: $(A\otimes B)(C\otimes D)=(AC)\otimes(BD)$

Moving beyond one qubit

Controlled-NOT gate: Flip target qubit if control qubit is 1

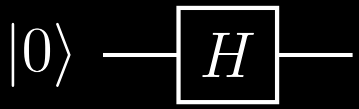


Input $(\boldsymbol{a}, \boldsymbol{b})$	Output $(\boldsymbol{a}, \boldsymbol{a} \oplus \boldsymbol{b})$
00	00
01	01
10	11
11	10

$$|10\rangle = |1\rangle \otimes |0\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

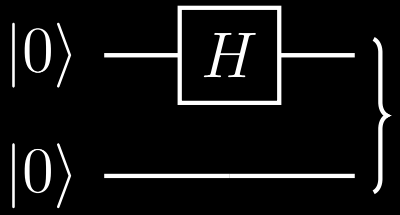
$$\mathrm{CNOT}\,|10\rangle = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \otimes \begin{bmatrix} 0 \\ 1 \end{bmatrix} = |11\rangle$$

Superposition + Linearity → Entanglement



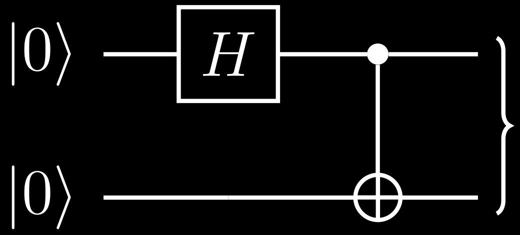
$$|+\rangle = \frac{|0\rangle+|1\rangle}{\sqrt{2}}$$

Superposition



$$|+\rangle \otimes |0\rangle = \frac{|00\rangle+|10\rangle}{\sqrt{2}}$$

Add a qubit



$$\text{CNOT}\left(\frac{|00\rangle+|10\rangle}{\sqrt{2}}\right) = \frac{|00\rangle+|11\rangle}{\sqrt{2}}$$

Linearity

Entanglement!

$\frac{|00\rangle + |11\rangle}{\sqrt{2}}$ CANNOT be expressed as a tensor product $|\psi\rangle \otimes |\phi\rangle$

The multi-qubit formalism

- Computational basis states:

$$|\mathsf{v}\rangle = |v_1 v_2 \cdots v_n\rangle = |v_1\rangle \otimes |v_2\rangle \otimes \cdots \otimes |v_n\rangle \in \mathbb{C}^{2^n};\ v_i \in \{0,1\}$$

- State vector for an n-qubit state:

$$|\psi\rangle = \sum_{\mathsf{v}\in\{0,1\}^n} \alpha_{\mathsf{v}} |\mathsf{v}\rangle \in \mathbb{C}^{2^n}\ ;\ \alpha_{\mathsf{v}} \in \mathbb{C},\ \||\psi\rangle\|_2^2 = \sum_{\mathsf{v}\in\{0,1\}^n} |\alpha_{\mathsf{v}}|^2 = 1$$

- Unitary operations on the state:

$$|\psi\rangle \mapsto U|\psi\rangle \in \mathbb{C}^{2^n}\ ;\ U \in \mathbb{U}^{2^n \times 2^n}, U^{-1} = U^\dagger, \|U|\psi\rangle\|_2 = 1$$

- Projective measurement of an “observable” O:

$$O = O^\dagger = \sum_i m_i M_i\ ,\ \mathbb{P}[m_i] = \langle\psi| M_i |\psi\rangle\ ,\ |\psi_i\rangle = \frac{M_i|\psi\rangle}{\sqrt{\mathbb{P}[m_i]}}$$

Poll Question 15

What is the result of $(X \otimes Z)(|0\rangle \otimes |1\rangle)$?

$A.$ $|1\rangle \otimes |1\rangle$

$B.$ $-|1\rangle \otimes |0\rangle$

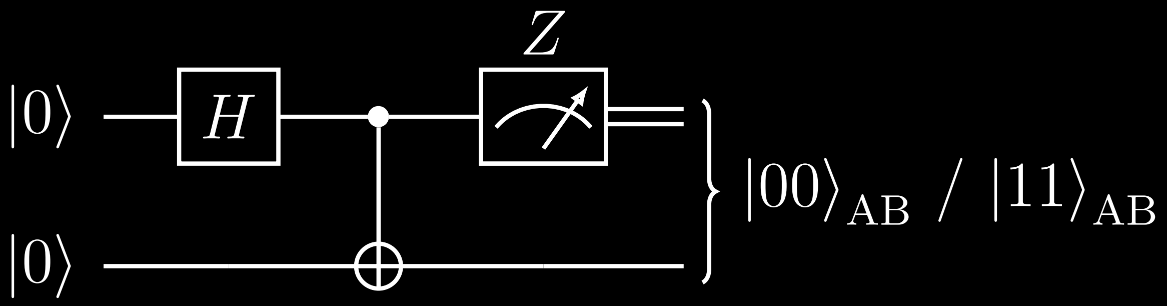
$C.$ $-|0\rangle \otimes |1\rangle$

$D.$ $-|1\rangle \otimes |1\rangle$

E. I'm not sure

Entanglement and Stabilizers

Entanglement: Bell Pairs



What happens if we measure the first qubit in the Z basis?

$$M_{+1} = |0\rangle\langle 0|_{\mathrm{A}} \otimes I_{\mathrm{B}} \,, \; M_{-1} = |1\rangle\langle 1|_{\mathrm{A}} \otimes I_{\mathrm{B}}$$

First qubit collapses to 0/1 & so does the second qubit too!

Bell Basis: $|\Phi^{\pm}\rangle_{\mathrm{AB}} = \dfrac{|00\rangle_{\mathrm{AB}} \pm |11\rangle_{\mathrm{AB}}}{\sqrt{2}} \,, \; |\Psi^{\pm}\rangle = \dfrac{|01\rangle_{\mathrm{AB}} \pm |10\rangle_{\mathrm{AB}}}{\sqrt{2}}$

Stabilizers of Bell Pairs

Bell Basis: $|\Phi^{\pm}\rangle_{\mathrm{AB}} = \dfrac{|00\rangle_{\mathrm{AB}} \pm |11\rangle_{\mathrm{AB}}}{\sqrt{2}}$, $|\Psi^{\pm}\rangle_{\mathrm{AB}} = \dfrac{|01\rangle_{\mathrm{AB}} \pm |10\rangle_{\mathrm{AB}}}{\sqrt{2}}$

These are ± 1-eigenvalued eigenvectors of $ZZ, XX, -YY, II$:

$$\begin{aligned} Z_{\mathrm{A}} Z_{\mathrm{B}} |\Phi^{\pm}\rangle_{\mathrm{AB}} &= (Z \otimes Z)\left(\frac{|0\rangle \otimes |0\rangle \pm |1\rangle \otimes |1\rangle}{\sqrt{2}}\right) \\ &= \frac{Z|0\rangle \otimes Z|0\rangle \pm Z|1\rangle \otimes Z|1\rangle}{\sqrt{2}} \\ &= \frac{|0\rangle \otimes |0\rangle \pm (-|1\rangle) \otimes (-|1\rangle)}{\sqrt{2}} \\ &= |\Phi^{\pm}\rangle_{\mathrm{AB}} \end{aligned}$$

Stabilizers of Bell Pairs

Bell Basis: $|\Phi^{\pm}\rangle_{\mathrm{AB}} = \frac{|00\rangle_{\mathrm{AB}} \pm |11\rangle_{\mathrm{AB}}}{\sqrt{2}}$, $|\Psi^{\pm}\rangle_{\mathrm{AB}} = \frac{|01\rangle_{\mathrm{AB}} \pm |10\rangle_{\mathrm{AB}}}{\sqrt{2}}$

These are $\pm$ 1-eigenvalued eigenvectors of $ZZ, XX, -YY, II$:

$$\begin{aligned}
Z_{\mathrm{A}} Z_{\mathrm{B}} |\Psi^{\pm}\rangle_{\mathrm{AB}} &= (Z \otimes Z) \left(\frac{|0\rangle \otimes |1\rangle \pm |1\rangle \otimes |0\rangle}{\sqrt{2}} \right) \\
&= \frac{Z|0\rangle \otimes Z|1\rangle \pm Z|1\rangle \otimes Z|0\rangle}{\sqrt{2}} \\
&= \frac{|0\rangle \otimes (-|1\rangle) \pm (-|1\rangle) \otimes |0\rangle}{\sqrt{2}} \\
&= -|\Psi^{\pm}\rangle_{\mathrm{AB}}
\end{aligned}$$

Stabilizer States

Bell Basis: $|\Phi^{\pm}\rangle_{\mathrm{AB}} = \dfrac{|00\rangle_{\mathrm{AB}} \pm |11\rangle_{\mathrm{AB}}}{\sqrt{2}}$, $|\Psi^{\pm}\rangle_{\mathrm{AB}} = \dfrac{|01\rangle_{\mathrm{AB}} \pm |10\rangle_{\mathrm{AB}}}{\sqrt{2}}$

(EPR: Einstein-Podolsky-Rosen)

$\updownarrow$ $\updownarrow$

Stabilizers: $\langle Z_{\mathrm{A}} Z_{\mathrm{B}} \, , \, \pm X_{\mathrm{A}} X_{\mathrm{B}} \rangle$ $\langle -Z_{\mathrm{A}} Z_{\mathrm{B}} \, , \, \pm X_{\mathrm{A}} X_{\mathrm{B}} \rangle$

Elements of the stabilizer must **mutually commute** to have a common eigenbasis, i.e., the same set of eigenvectors diagonalize all stabilizer elements. **Key fact:** $XZ = -ZX$

An n-qubit stabilizer state has n Pauli stabilizer generators

Stabilizer States

Bell Basis: $|\Phi^{\pm}\rangle_{\mathrm{AB}} = \dfrac{|00\rangle_{\mathrm{AB}} \pm |11\rangle_{\mathrm{AB}}}{\sqrt{2}}$, $|\Psi^{\pm}\rangle_{\mathrm{AB}} = \dfrac{|01\rangle_{\mathrm{AB}} \pm |10\rangle_{\mathrm{AB}}}{\sqrt{2}}$

(EPR: Einstein-Podolsky-Rosen)

Stabilizers: $\langle Z_{\mathrm{A}} Z_{\mathrm{B}} \; , \; \pm X_{\mathrm{A}} X_{\mathrm{B}} \rangle$ $\langle - Z_{\mathrm{A}} Z_{\mathrm{B}} \; , \; \pm X_{\mathrm{A}} X_{\mathrm{B}} \rangle$

GHZ Basis: $|\mathrm{GHZ}\rangle_{\mathrm{ABC}} = \dfrac{|000\rangle_{\mathrm{ABC}} + |111\rangle_{\mathrm{ABC}}}{\sqrt{2}}$ and its variants

(GHZ: Greenberger-Horne-Zeilinger)

Stabilizers: $\langle \pm Z_{\mathrm{A}} Z_{\mathrm{B}} I_{\mathrm{C}} \; , \; \pm I_{\mathrm{A}} Z_{\mathrm{B}} Z_{\mathrm{C}} \; , \; \pm X_{\mathrm{A}} X_{\mathrm{B}} X_{\mathrm{C}} \rangle$

An n-qubit stabilizer state has n Pauli stabilizer generators

Poll Question 16

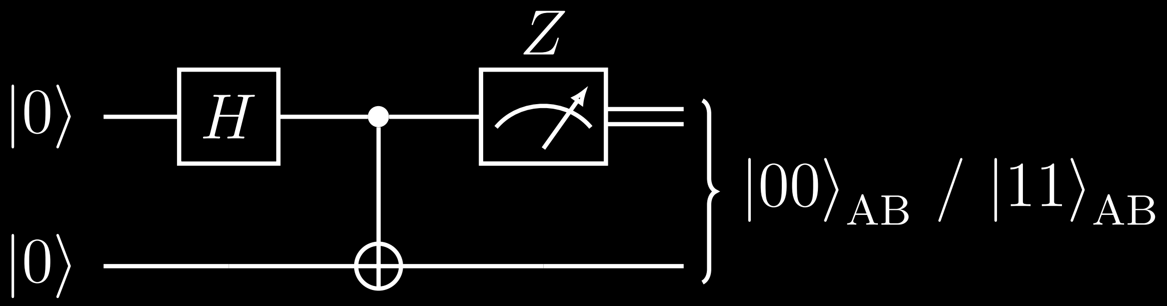
What are the stabilizer generators of $\frac{|001\rangle + |110\rangle}{\sqrt{2}}$? The state must have eigenvalue $+1$ for these operators.

A. $\langle ZZI, IZZ, XXX \rangle$
B. $\langle ZZI, IZZ, -XXX \rangle$
C. $\langle ZZI, -IZZ, XXX \rangle$
D. $\langle -ZZI, IZZ, XXX \rangle$
E. $\langle ZZI, -IZZ, -XXX \rangle$
F. $\langle -ZZI, -IZZ, XXX \rangle$

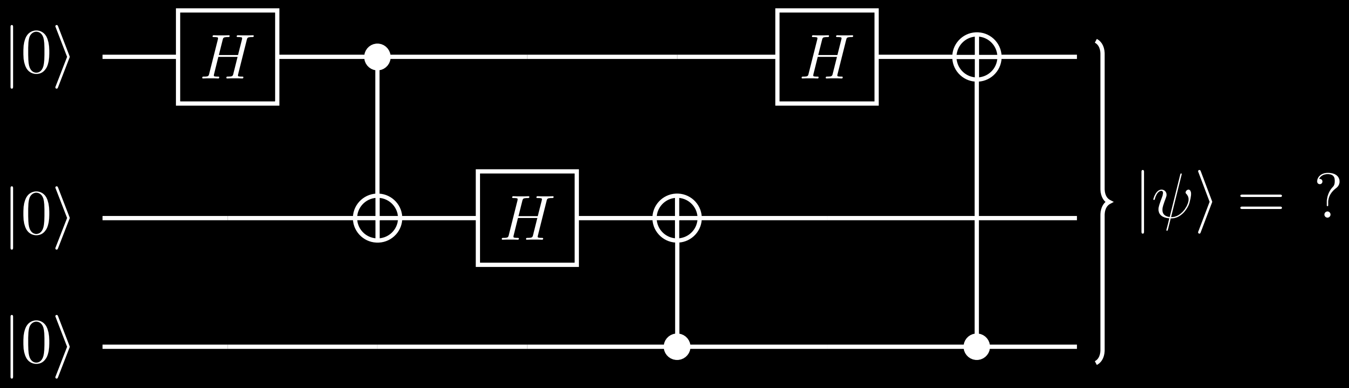
G. I'm not sure

The Stabilizer Formalism

Operations on stabilizer states



Here we can track the state quite easily, with length 4 vectors



This is more complicated to track! With n qubits we have length 2^n!

Clifford gates

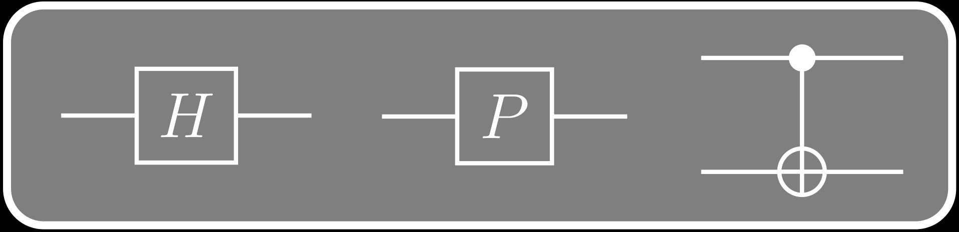
Unitary operations U that map Paulis to Paulis under conjugation

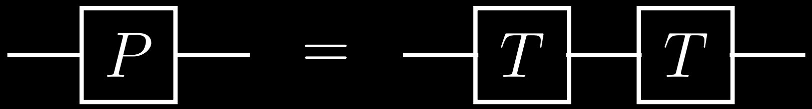
$$\mathcal{C}_n = \{U \in \mathbb{U}^{2^n} : UEU^\dagger = E' \in \mathcal{P}_n \text{ for all } E \in \mathcal{P}_n\}$$

$$\mathcal{P}_n = \{\pm\imath E_1 \otimes E_2 \otimes \cdots \otimes E_n \; ; \; E_j \in \{I, X, Y, Z\}, j = 1, 2, \ldots, n\}$$

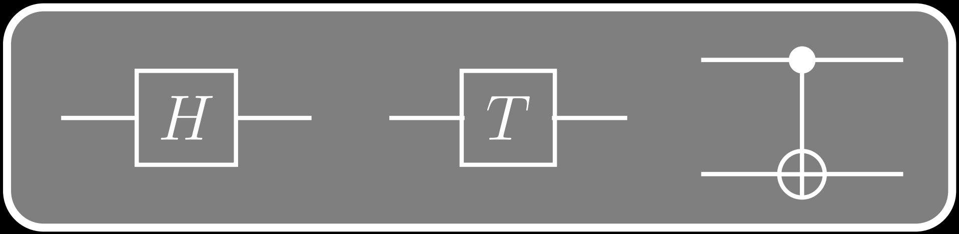
(Clifford group & Pauli group)

$\mathcal{C}_n =$ n-qubit Clifford gates:

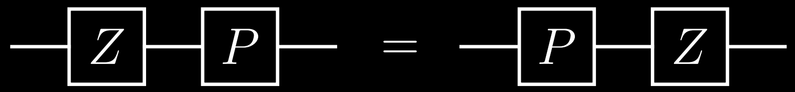




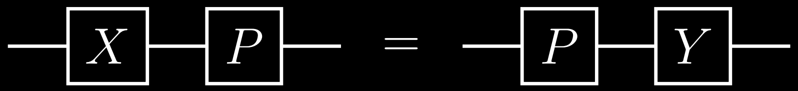
Universality: “Clifford + T”

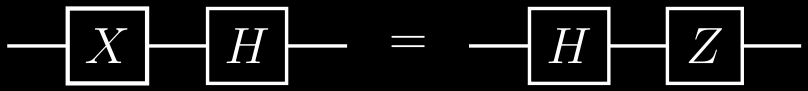


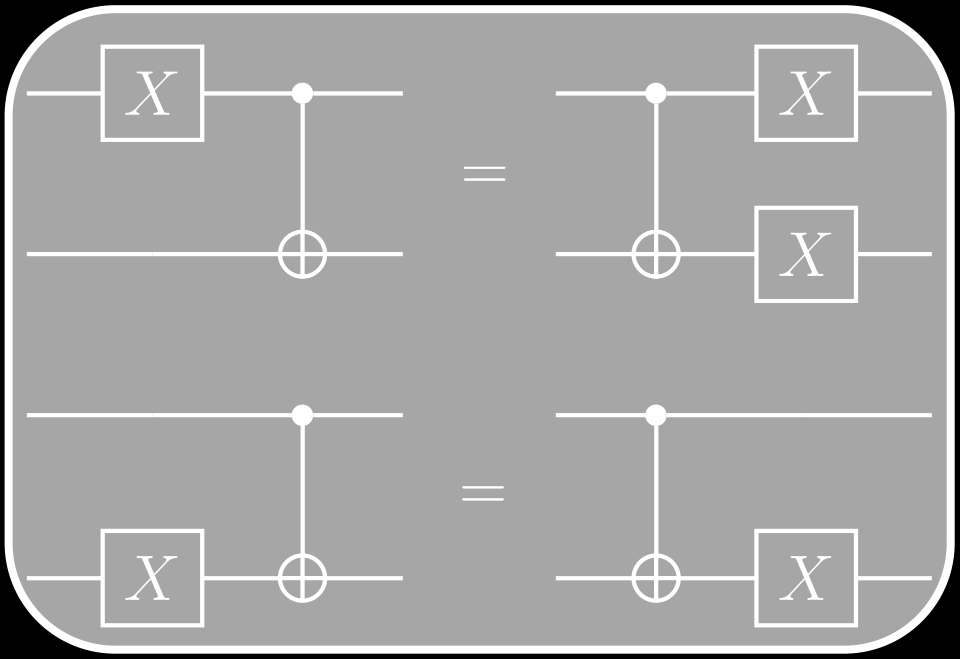
Clifford gates: Pauli tracking

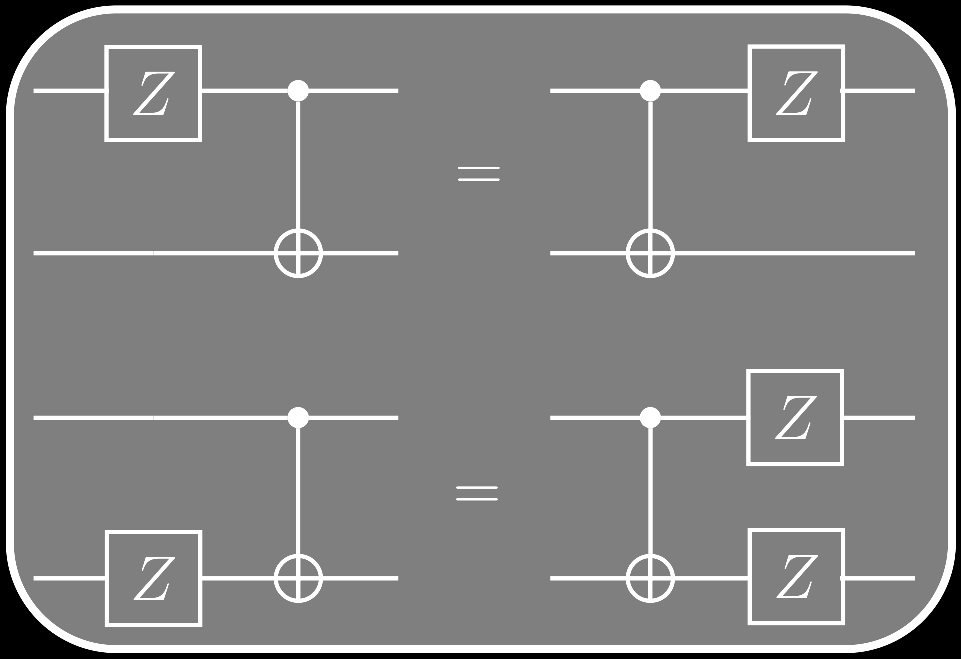


$$\boxed{\mathcal{C}_n : UEU^\dagger = E' \iff UE = E'U}$$

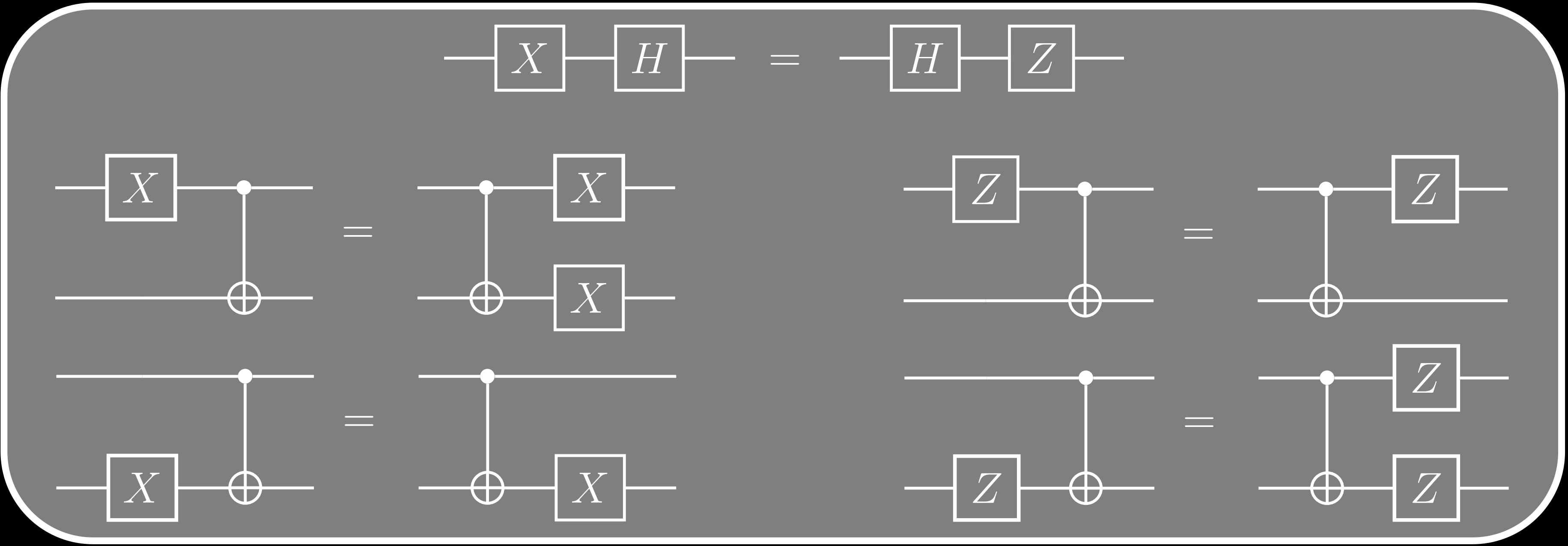


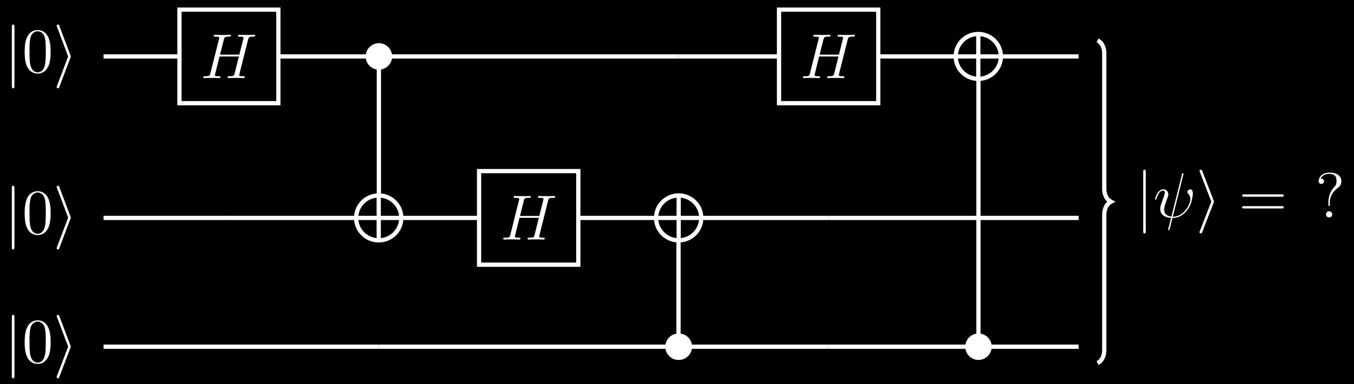




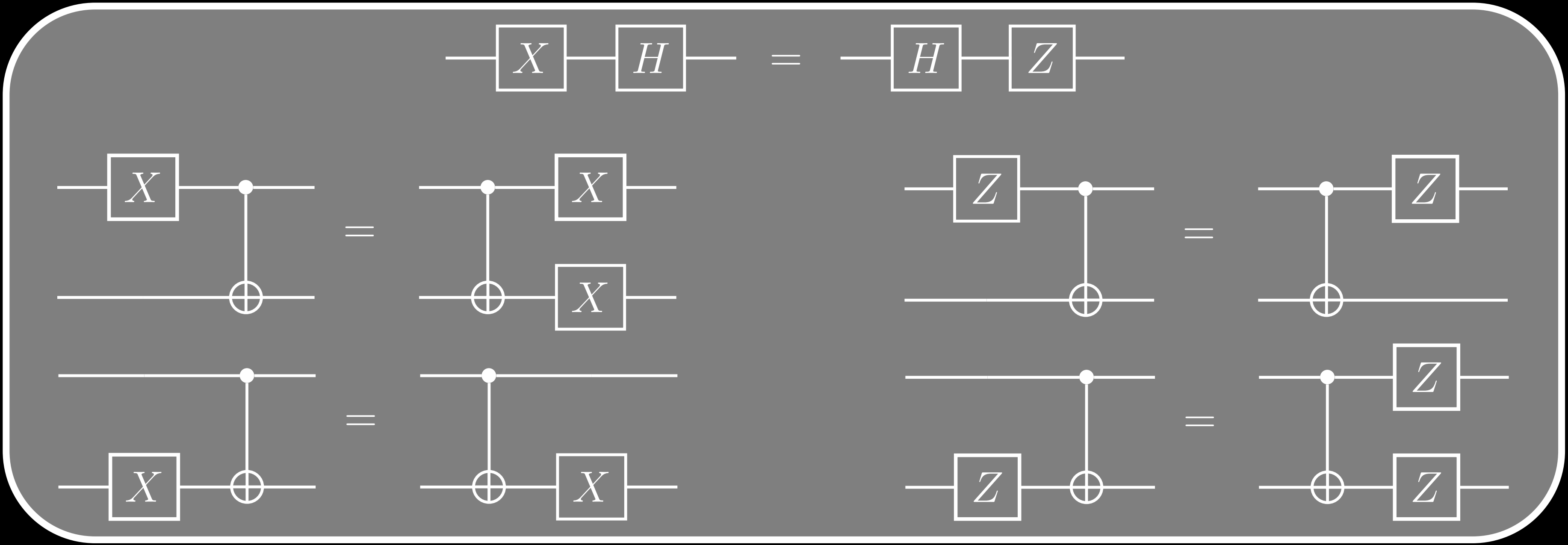


Clifford gates: Pauli tracking

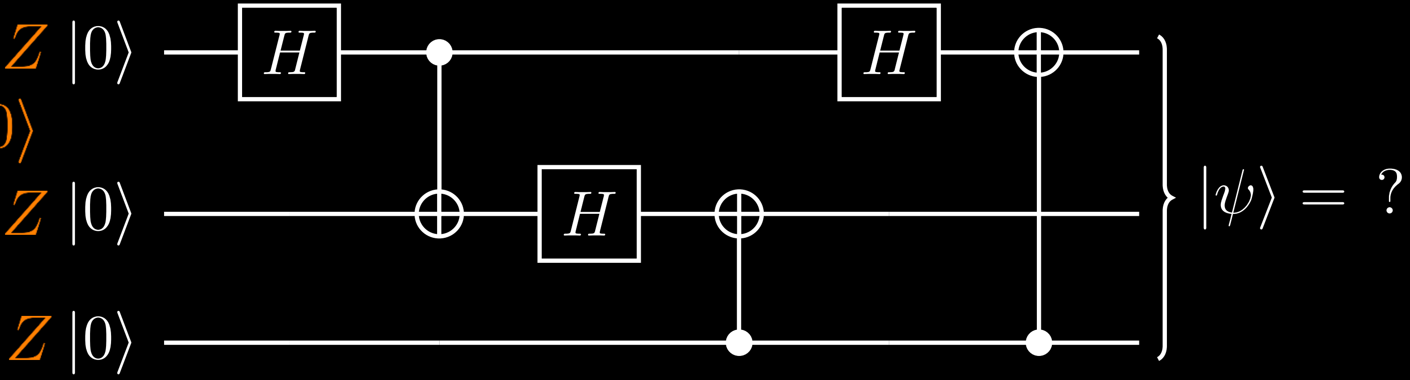




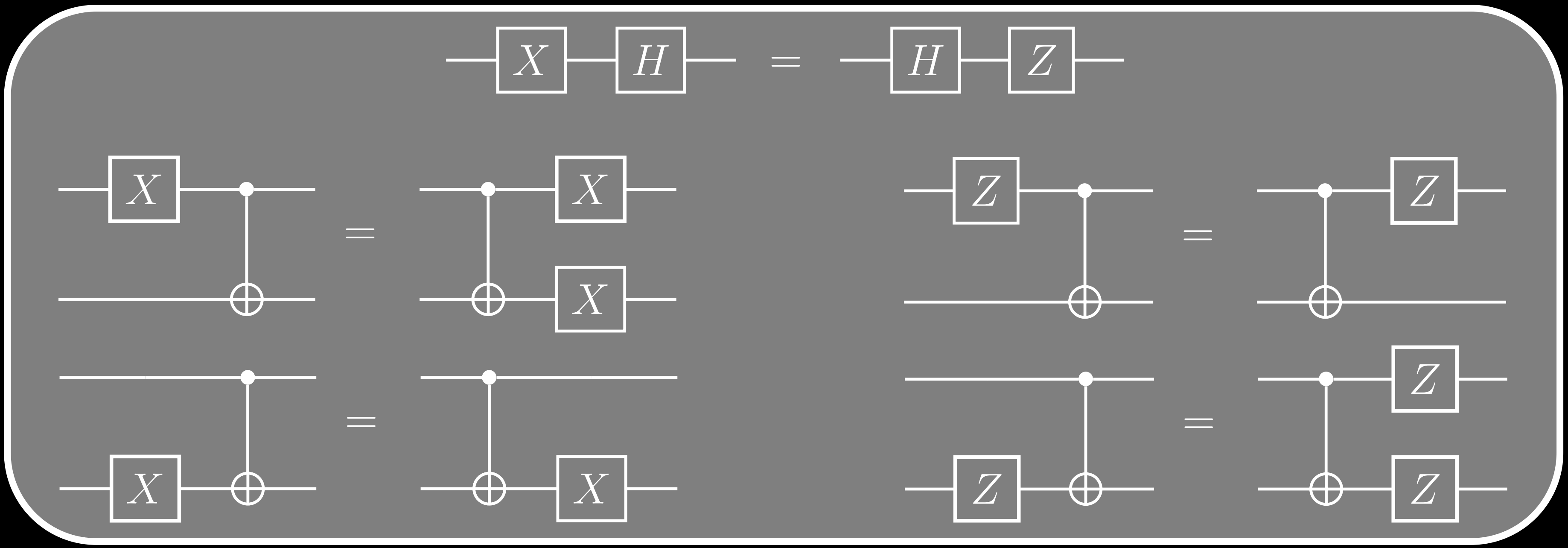
Clifford gates: Pauli tracking

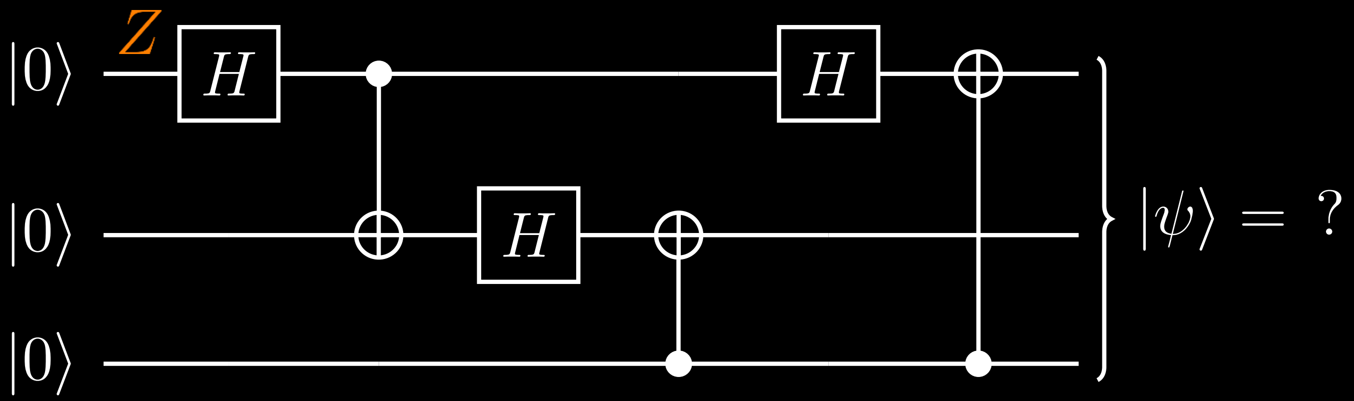


$Z\,|0\rangle = |0\rangle$

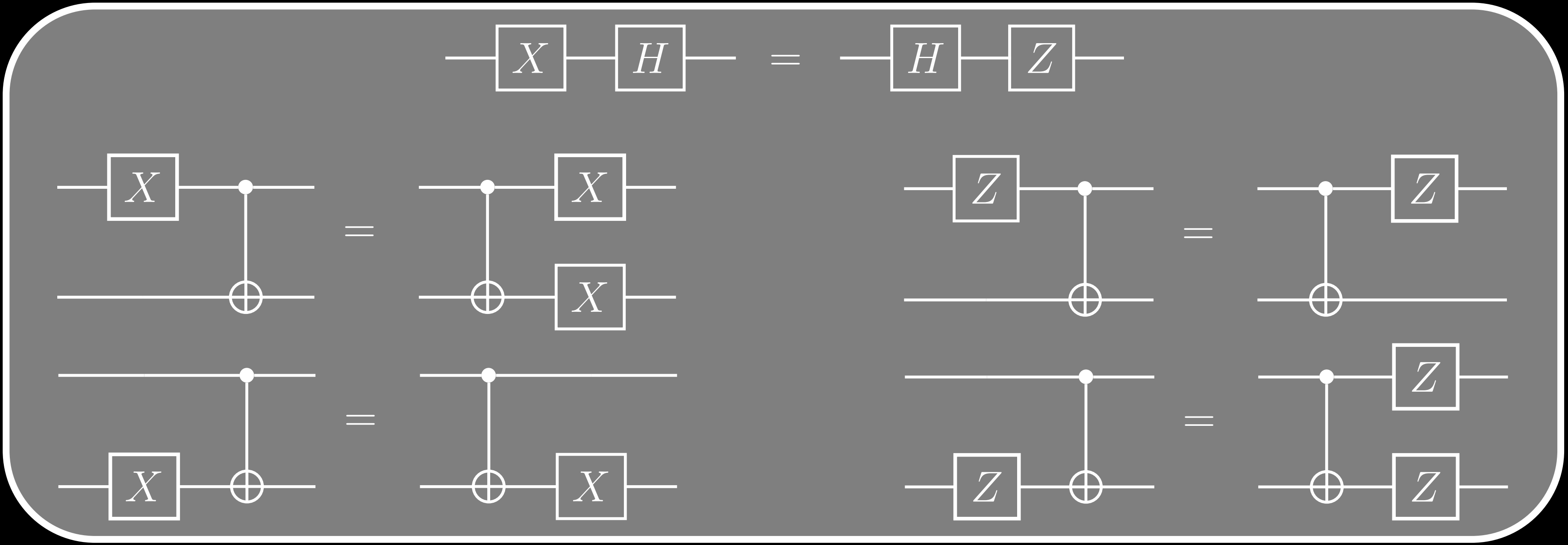


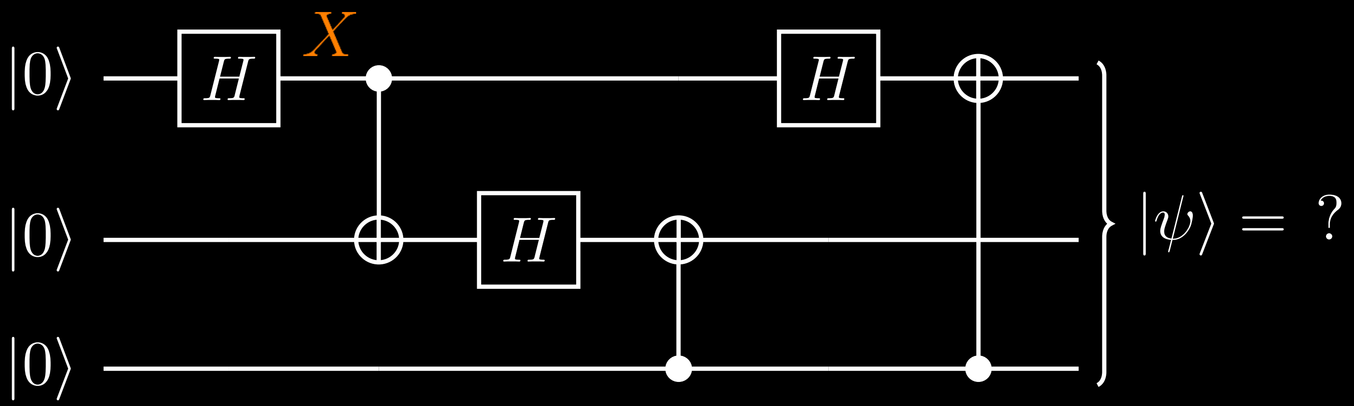
Clifford gates: Pauli tracking



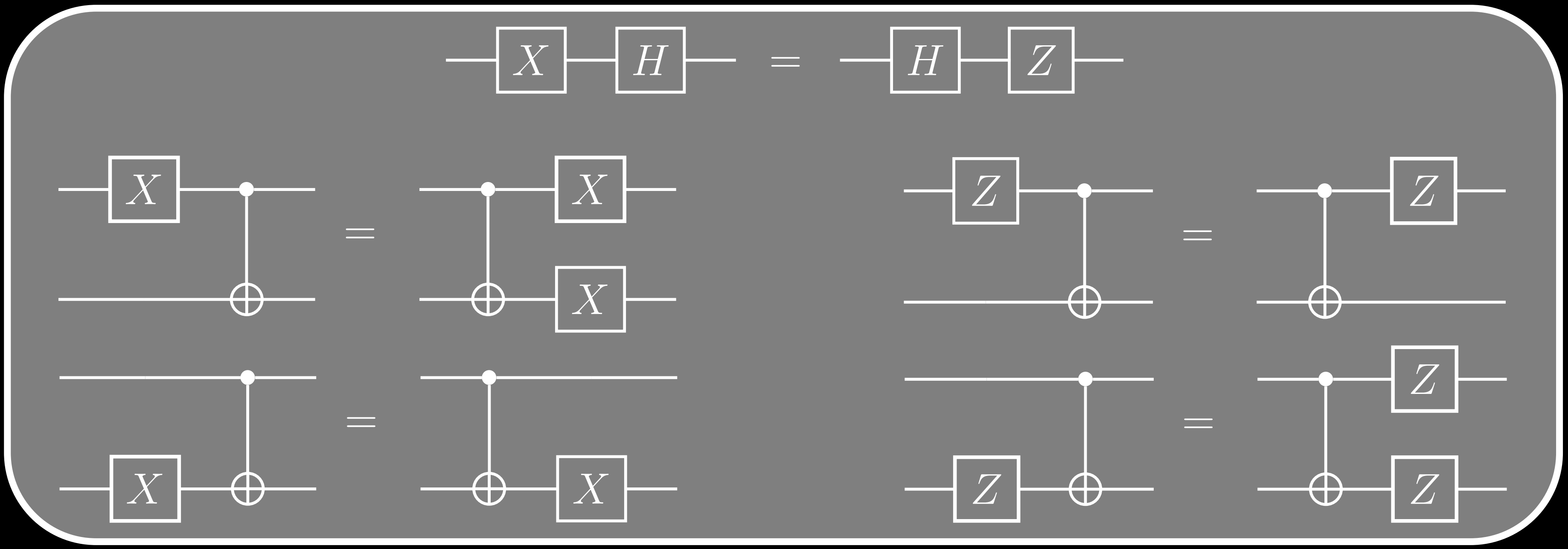


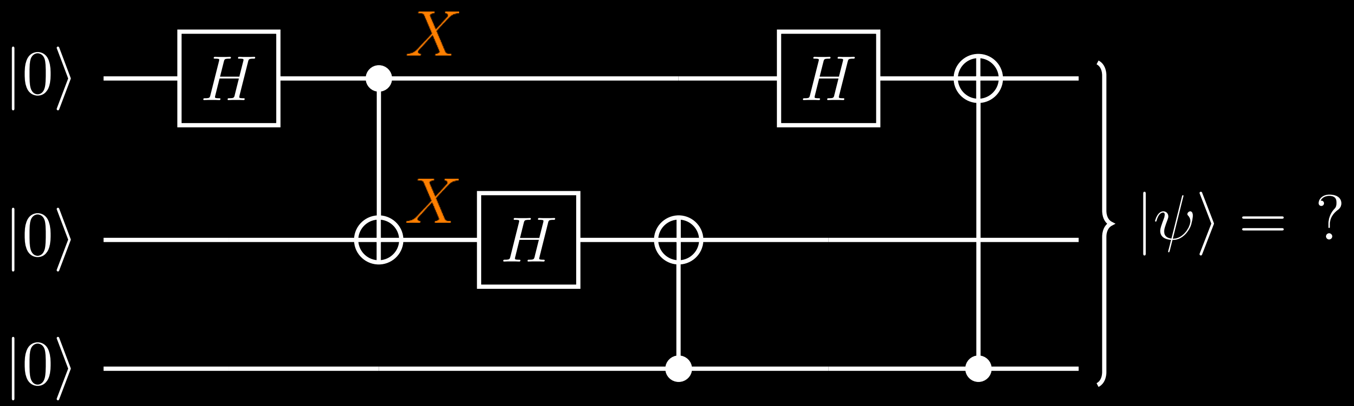
Clifford gates: Pauli tracking



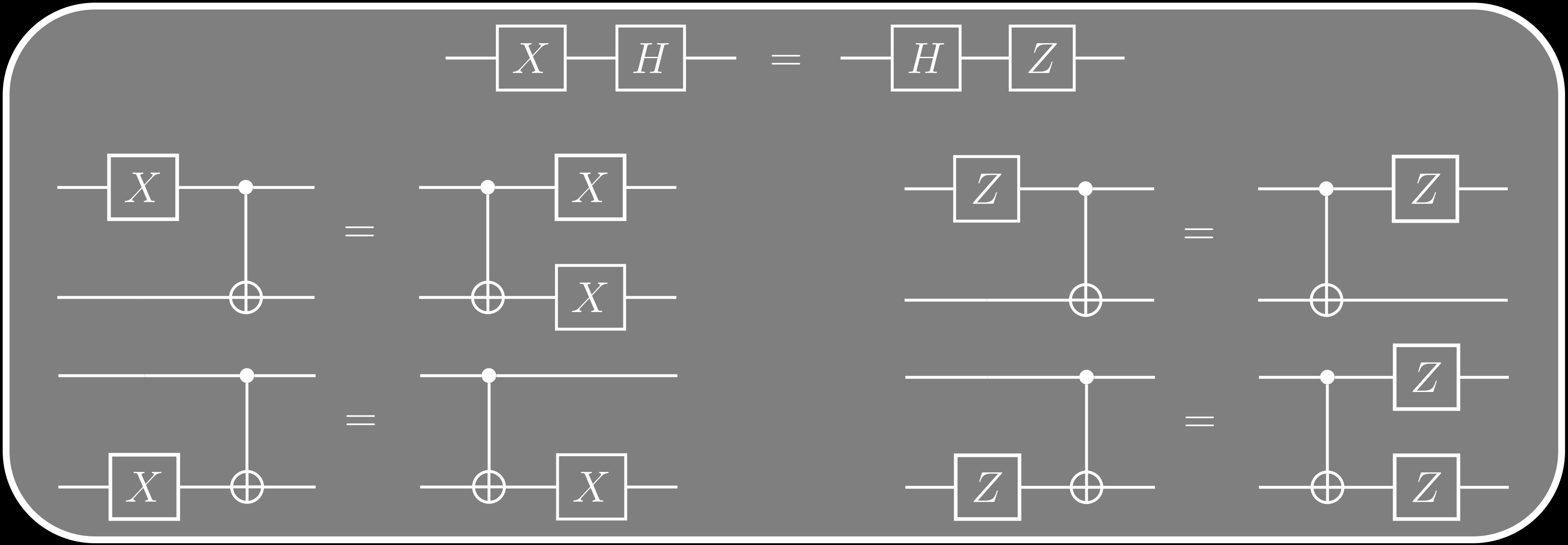


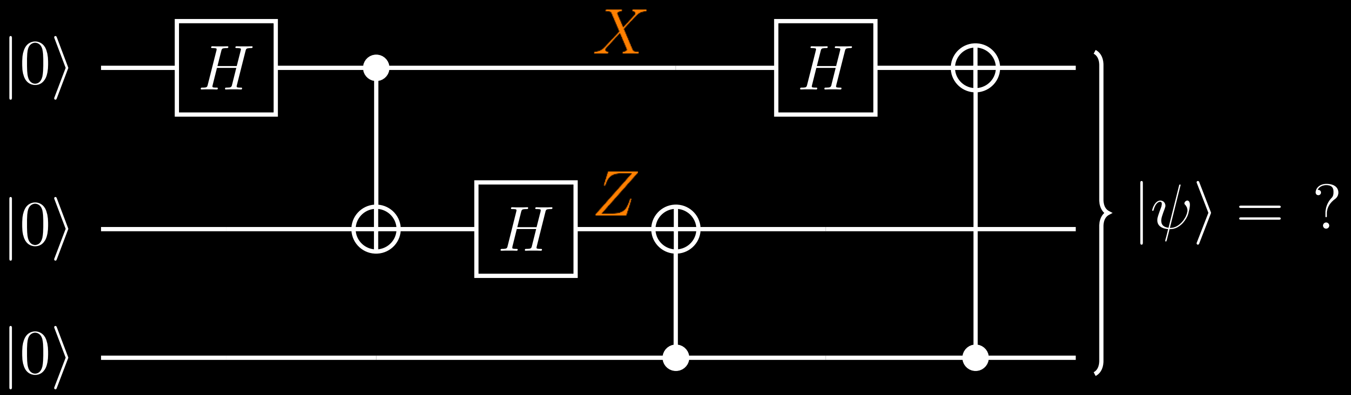
Clifford gates: Pauli tracking



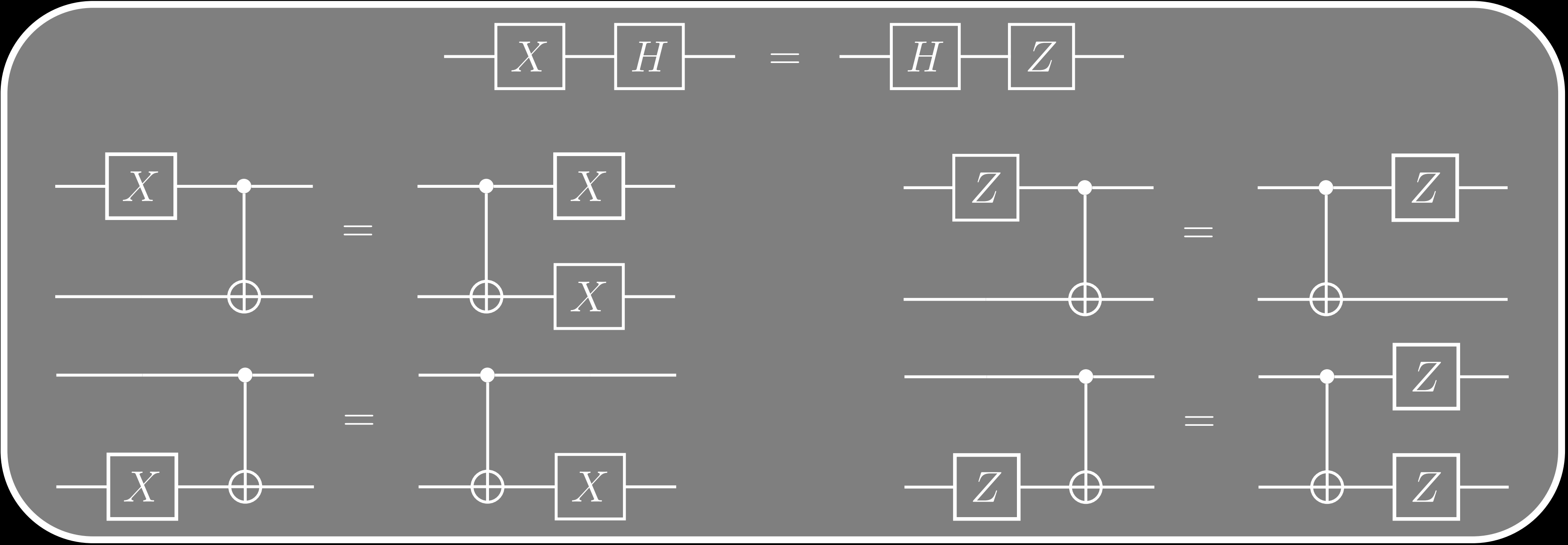


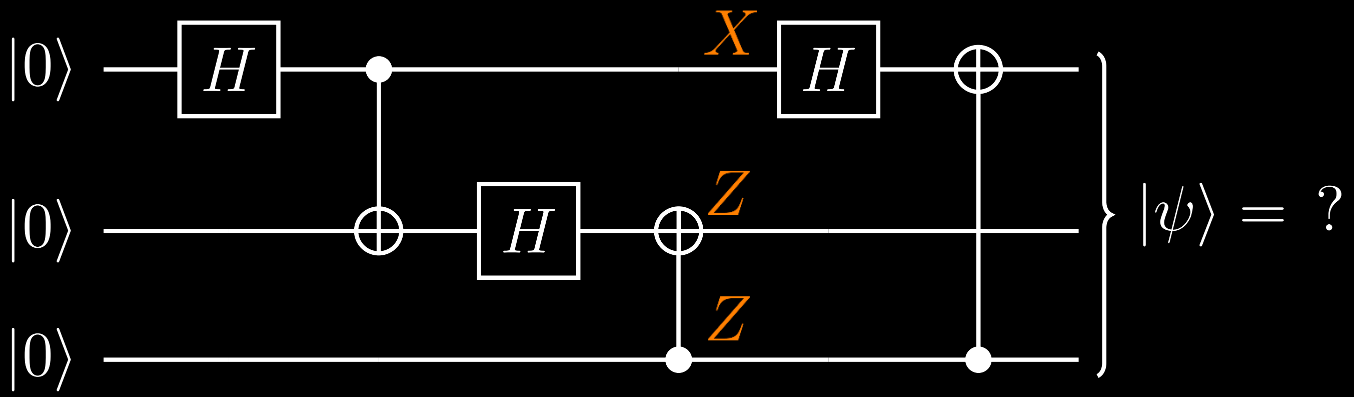
Clifford gates: Pauli tracking



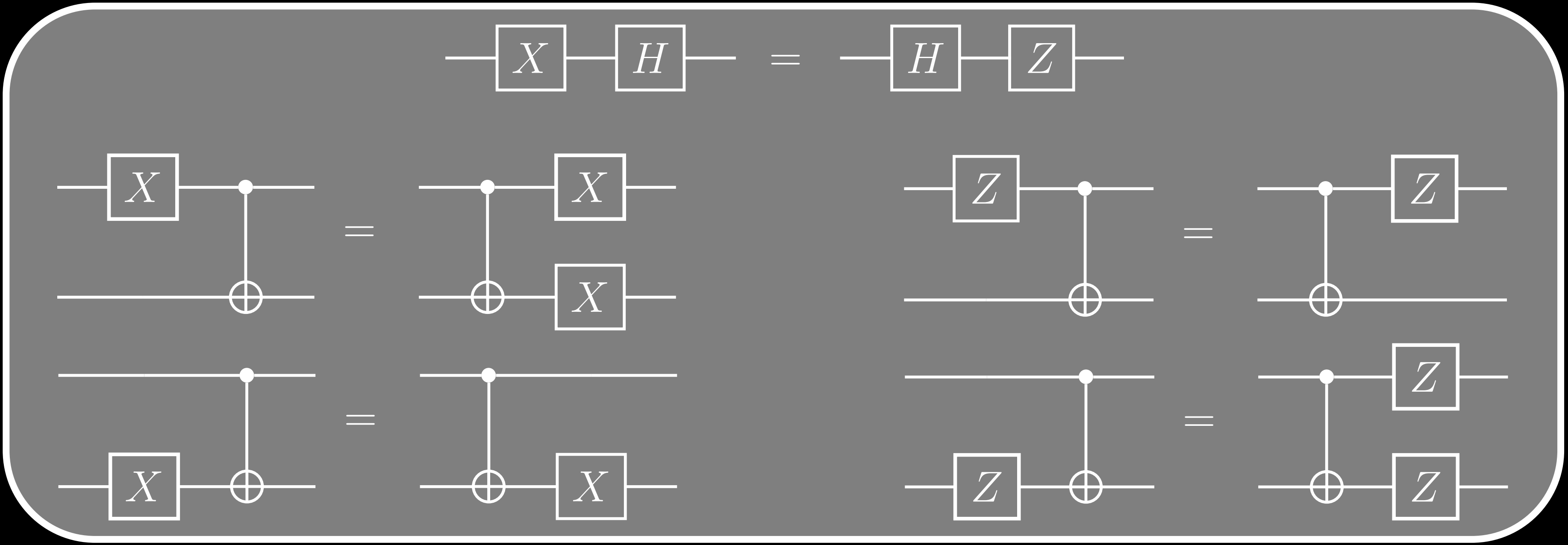


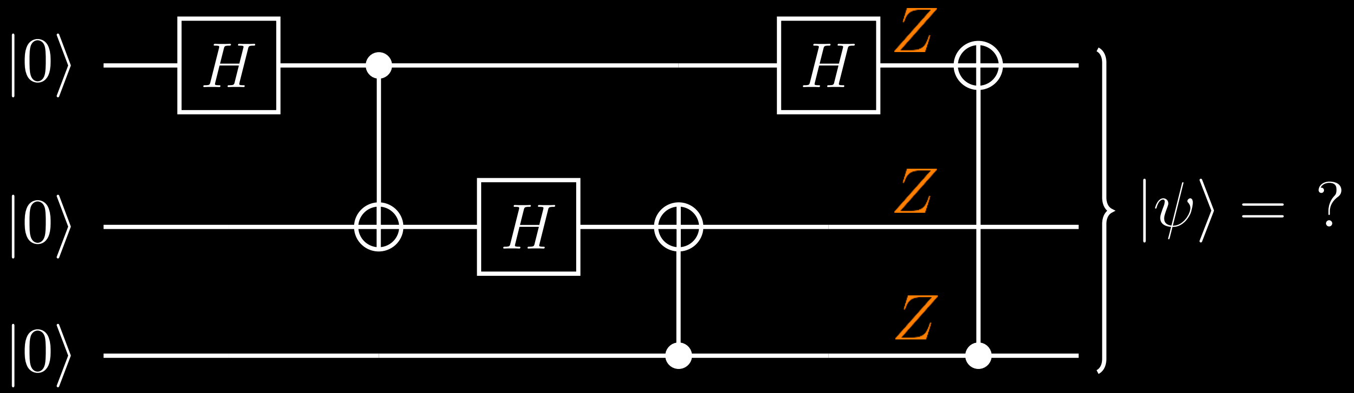
Clifford gates: Pauli tracking



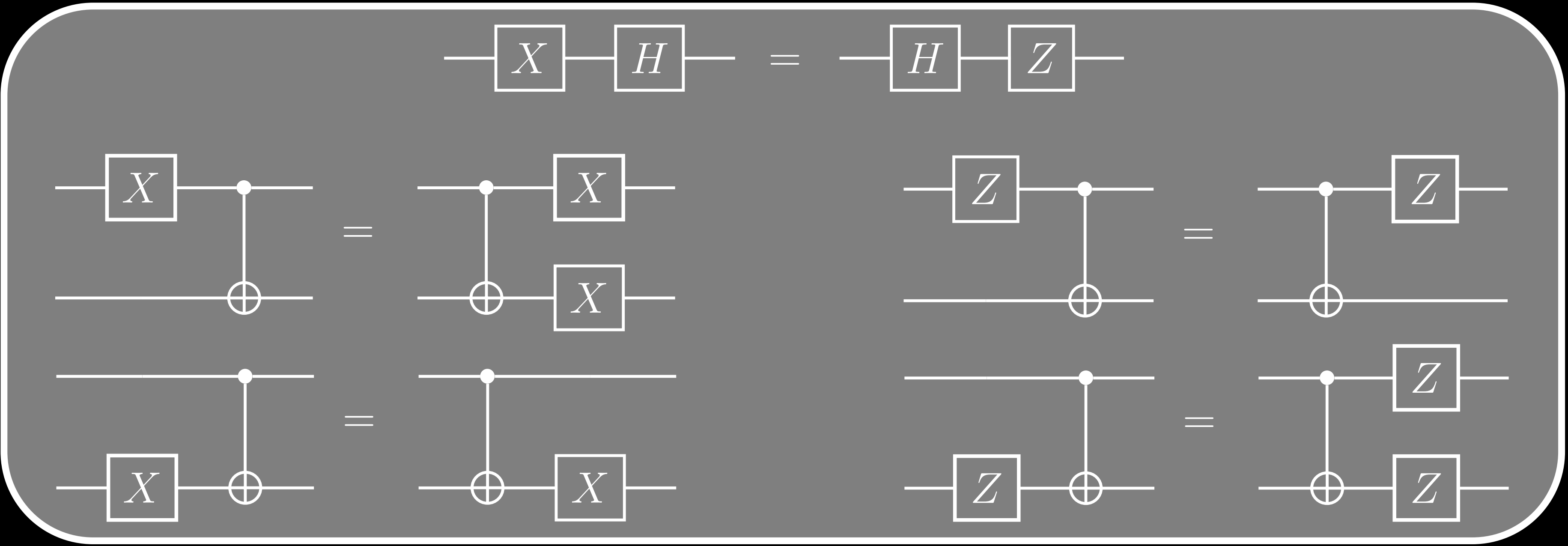


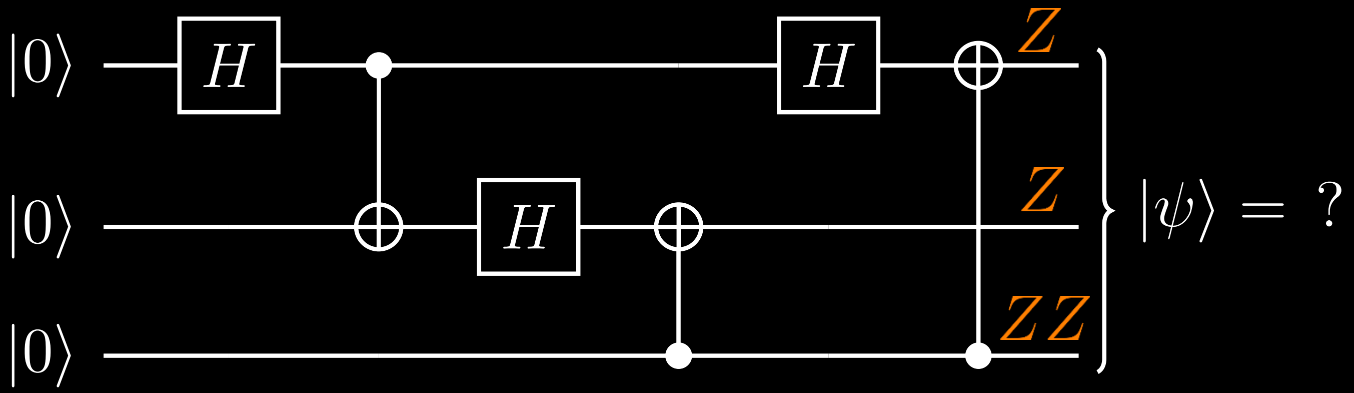
Clifford gates: Pauli tracking



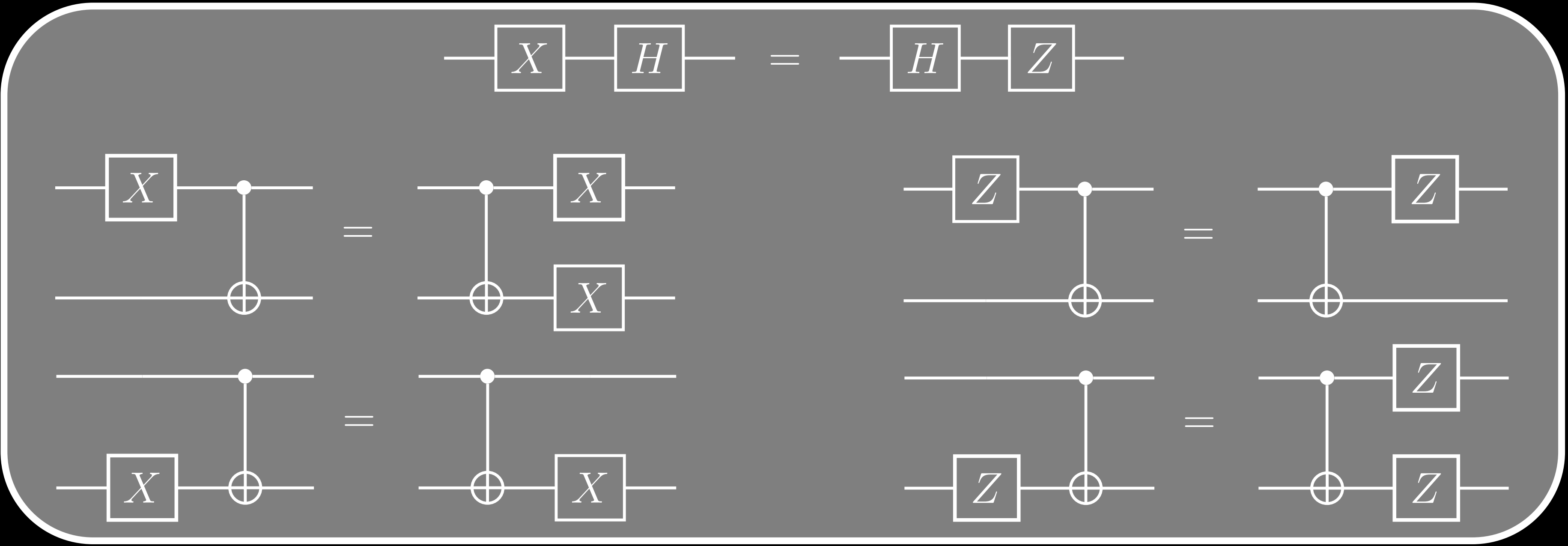


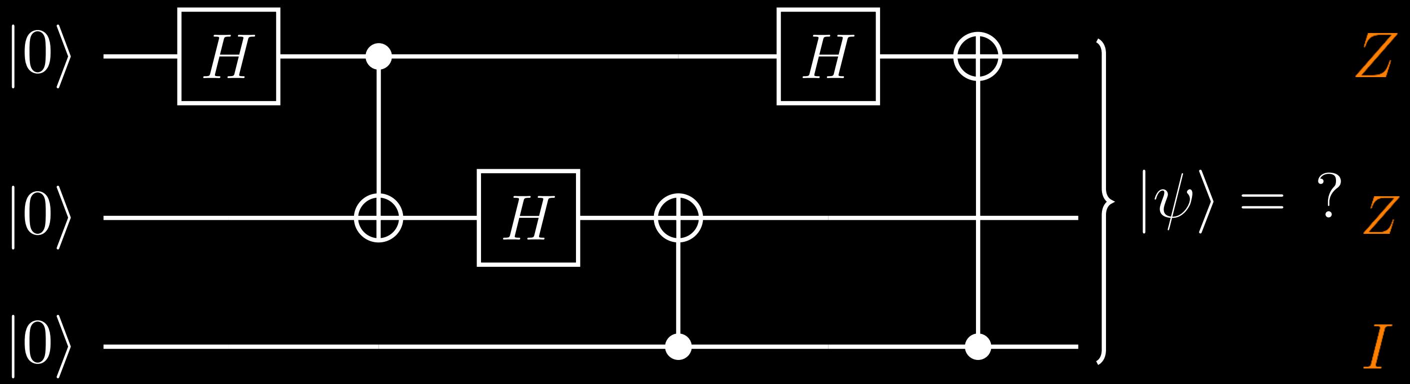
Clifford gates: Pauli tracking



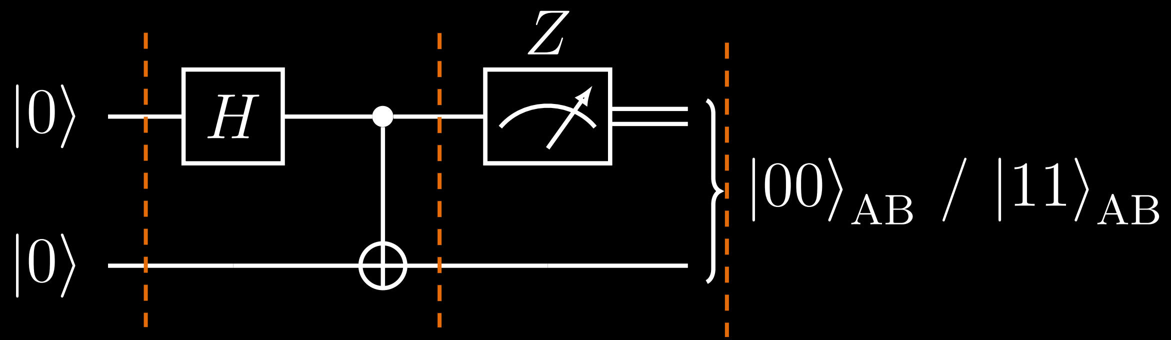


Clifford gates: Pauli tracking





Pauli measurements



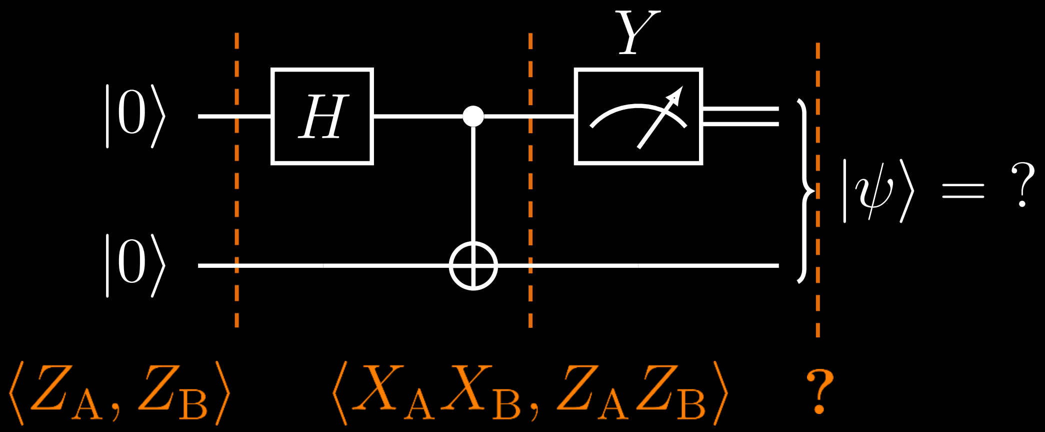
$\langle Z_{\mathrm{A}}, Z_{\mathrm{B}}\rangle$ $\langle X_{\mathrm{A}}X_{\mathrm{B}}, Z_{\mathrm{A}}Z_{\mathrm{B}}\rangle$? $XZ = -ZX$

Compare measured operator Z_A with each stabilizer

1. Z_A anticommutes with $X_A\, X_B$: replace with $\pm Z_A$
2. Z_A commutes with $Z_A\, Z_B$: retain the stabilizer

Output stabilizer: $\langle \pm Z_A\,, Z_A Z_B\rangle \equiv \langle \pm Z_A\,, \pm Z_B\rangle \equiv |00\rangle/|11\rangle$

Pauli measurements


$|0\rangle$
$|0\rangle$
H
Y
$|\psi\rangle = ?$
$\langle Z_\mathrm{A}, Z_\mathrm{B}\rangle$
$\langle X_\mathrm{A}X_\mathrm{B}, Z_\mathrm{A}Z_\mathrm{B}\rangle$
?

$$XZ = -ZX$$
$$XY = -YX$$
$$ZY = -YZ$$

Compare measured operator Y_A with each stabilizer

1. Y_A anticommutes with $X_A\, X_B$: replace with $\pm\, Y_A$

2. Y_A anticommutes with $Z_A\, Z_B$: multiply $Z_A\, Z_B$ with $X_A\, X_B$

Output stabilizer: $\langle \pm\, Y_A\, , -Y_AY_B\rangle \equiv \langle \pm\, Y_A\, , \mp\, Y_B\rangle$

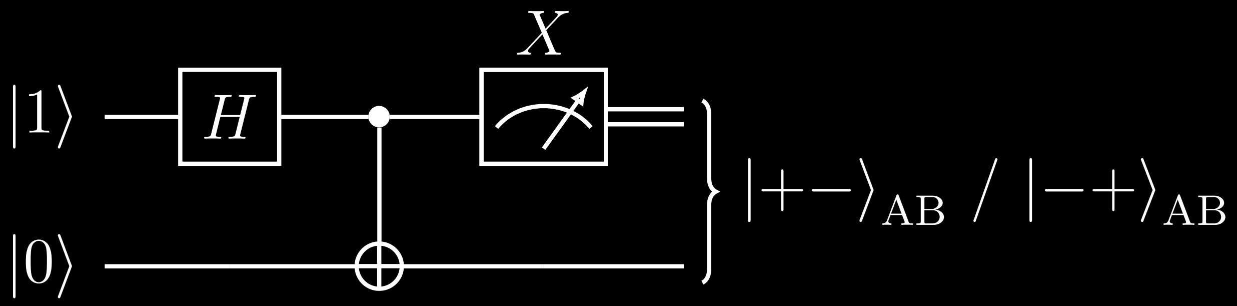
Gottesman-Knill theorem

Clifford gates and Pauli measurements on input stabilizer states can be efficiently simulated classically, by simply tracking the stabilizers of the input state through the circuit!

Stabilizer Circuits: Cliffords + Pauli measurements

Poll Question 17

What are the stabilizers for the output of the following circuit if the measurement result is +1?



X
$|1\rangle$
H
$|0\rangle$
$|{+}{-}\rangle_{\mathrm{AB}}$ / $|{-}{+}\rangle_{\mathrm{AB}}$

A. $\langle -XX, XI\rangle = \langle -IX, XI\rangle$
B. $\langle -XX, -XI\rangle = \langle IX, -XI\rangle$
C. $\langle -XX, ZZ\rangle = \langle -XX, YY\rangle$
D. $\langle XX, -XI\rangle = \langle -IX, -XI\rangle$

E. I’m not sure

Protecting information with entanglement

Pauli errors (and beyond)

- Recall the n-qubit Pauli group:

$$\mathcal{P}_n = \{\pm\imath E_1 \otimes E_2 \otimes \cdots \otimes E_n \ ; \ E_j \in \{I, X, Y, Z\}, j = 1, 2, \ldots, n\}$$

- Each element can also be thought of as an error operator, since Pauli matrices form an orthogonal basis for all matrices under the trace inner product: $\langle A, B\rangle_{\mathrm{Tr}} := \mathrm{Tr}(A^\dagger B)$

- Key Result: if Pauli errors on t qubits can be corrected, then any linear combination of them can also be corrected

- Goal: design quantum codes that correct Pauli errors

Stabilizer States

Bell Basis: $|\Phi^{\pm}\rangle_{\mathrm{AB}} = \dfrac{|00\rangle_{\mathrm{AB}} \pm |11\rangle_{\mathrm{AB}}}{\sqrt{2}}$, $|\Psi^{\pm}\rangle_{\mathrm{AB}} = \dfrac{|01\rangle_{\mathrm{AB}} \pm |10\rangle_{\mathrm{AB}}}{\sqrt{2}}$

(EPR: Einstein-Podolsky-Rosen)

↕ ↕

Stabilizers: $\langle Z_{\mathrm{A}} Z_{\mathrm{B}} \, , \, \pm X_{\mathrm{A}} X_{\mathrm{B}} \rangle$ $\langle -Z_{\mathrm{A}} Z_{\mathrm{B}} \, , \, \pm X_{\mathrm{A}} X_{\mathrm{B}} \rangle$

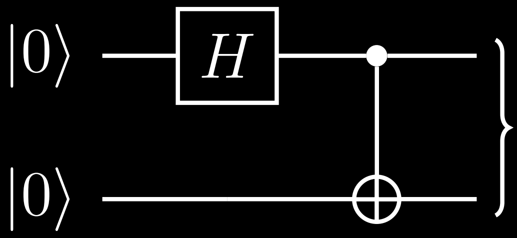
GHZ Basis: $|\mathrm{GHZ}\rangle_{\mathrm{ABC}} = \dfrac{|000\rangle_{\mathrm{ABC}} + |111\rangle_{\mathrm{ABC}}}{\sqrt{2}}$ and its variants

(GHZ: Greenberger-Horne-Zeilinger)

Stabilizers: $\langle \pm Z_{\mathrm{A}} Z_{\mathrm{B}} I_{\mathrm{C}} \, , \, \pm I_{\mathrm{A}} Z_{\mathrm{B}} Z_{\mathrm{C}} \, , \, \pm X_{\mathrm{A}} X_{\mathrm{B}} X_{\mathrm{C}} \rangle$

An n-qubit stabilizer state has n Pauli stabilizer generators

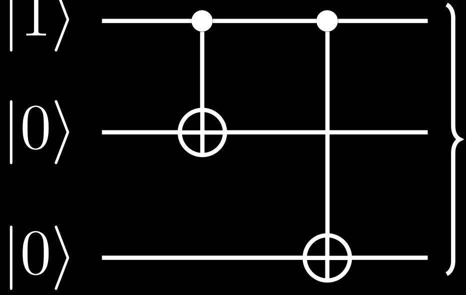
The three-qubit code



$$\text{CNOT}\left(\frac{|00\rangle+|10\rangle}{\sqrt{2}}\right)=\frac{|00\rangle+|11\rangle}{\sqrt{2}}$$

$$\langle Z_{\text{A}}Z_{\text{B}}\ ,\ X_{\text{A}}X_{\text{B}}\rangle$$

$$|\psi\rangle_L=\alpha\,|0\rangle+\beta\,|1\rangle$$



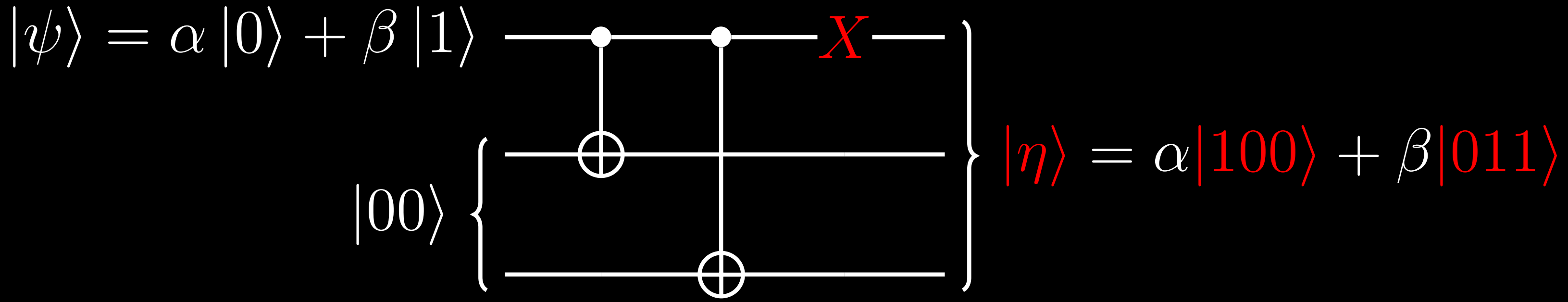
$$\overline{|\psi\rangle}=\overbrace{\alpha\,|000\rangle+\beta\,|111\rangle}^{\text{GHZ when } \alpha=\beta=\frac{1}{\sqrt{2}}}$$

From GHZ stabilizers $\langle ZZI, IZZ, XXX\rangle$ drop XXX to create a logical qubit!

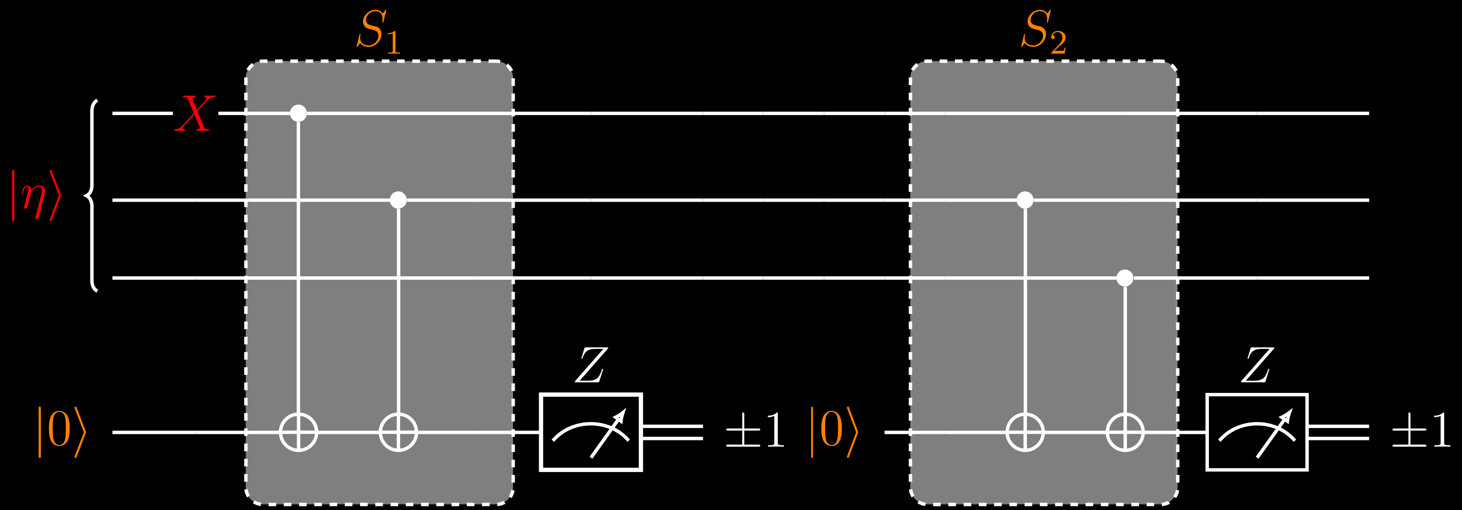
Stabilizers: $\langle ZZI, IZZ\rangle$ (they commute), $\overline{|\psi\rangle}$ is a +1-eigenvector for all α, β

Syndrome measurement

Suppose that after encoding the logical qubit the error X_1 acts on the state

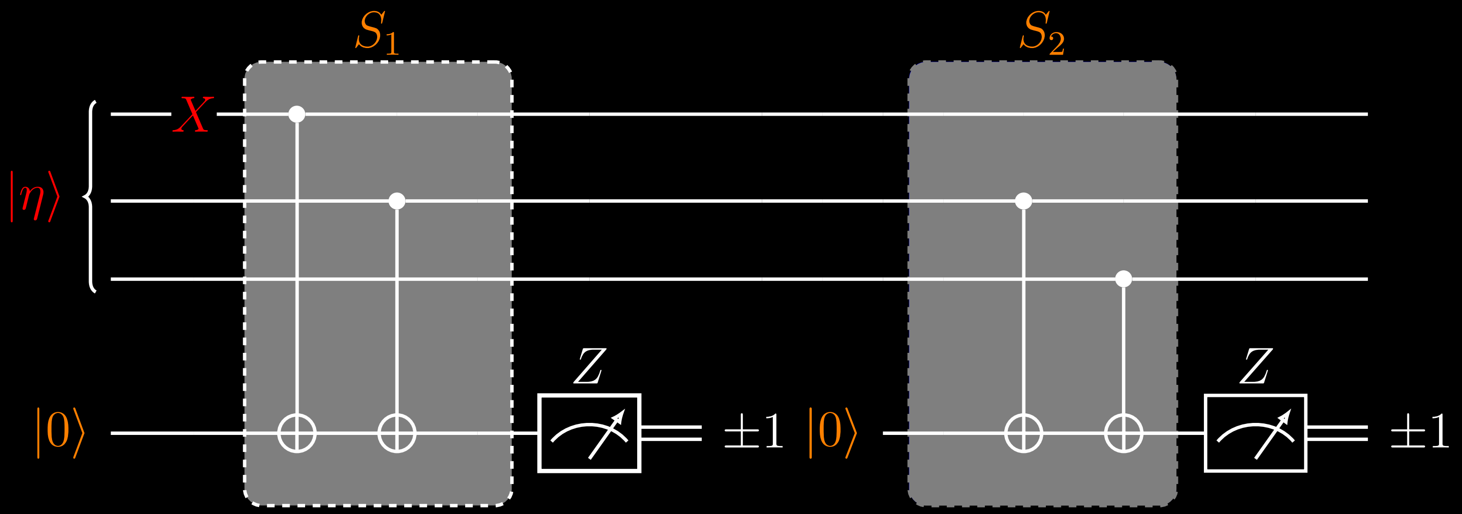

$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$
X
$|00\rangle$
$|\eta\rangle = \alpha|100\rangle + \beta|011\rangle$

Measure the stabilizer generators $S_1 = ZZI$ and $S_2 = IZZ$:



Syndrome measurement

Measure the stabilizer generators $S_1 = ZZI$ and $S_2 = IZZ$:



The error X propagates through the CNOT and flips the measurement

Hence, the measurement results in -1 whenever there are an odd number of X's on the ancilla (through the CNOT gates), i.e., **when the error anticommutes with the stabilizer S_i**

Poll Question 18

Given the stabilizers $\langle S_1 = ZZI, S_2 = IZZ \rangle$ of the code, what is the syndrome for the error IXI?

$A.$ $(+1, +1)$
$B.$ $(+1, -1)$
$C.$ $(-1, +1)$
$D.$ $(-1, -1)$

E. I'm not sure

Binary representation

Map an n-qubit Hermitian Pauli matrix to a pair of binary vectors:

Example for $n = 3$: $X \otimes Z \otimes Y \Longrightarrow E(\boldsymbol{a}, \boldsymbol{b})$

$\boldsymbol{a} = [1 \quad 0 \quad 1]$ (X component)

$\boldsymbol{b} = [0 \quad 1 \quad 0]$ (Z component)

How to check if $E(\boldsymbol{a}, \boldsymbol{b}) = X \otimes Z \otimes Y$ and $E(\boldsymbol{c}, \boldsymbol{d}) = Z \otimes Z \otimes X$ commute?

Compare operators on each qubit:

$$X \otimes Z \otimes Y \mapsto ([1\ 0\ 1], [0\ 1\ 0])$$

$$Z \otimes Z \otimes X \mapsto ([0\ 0\ 1], [1\ 1\ 0])$$

Symplectic inner product: $\langle [\boldsymbol{a}, \boldsymbol{b}], [\boldsymbol{c}, \boldsymbol{d}] \rangle_{\text{sym}} \coloneqq \boldsymbol{a}\boldsymbol{d}^T + \boldsymbol{b}\boldsymbol{c}^T$ (modulo 2)

$$= \begin{cases} 0 & \text{iff they commute,} \\ 1 & \text{iff they anticommute} \end{cases}$$

Binary representation: errors

Stabilizers ($n = 3$): $Z \otimes Z \otimes I$ $\quad$ $I \otimes Z \otimes Z$

(X component) $\boldsymbol{a_1} = [0 \quad 0 \quad 0]$ $\quad$ $\boldsymbol{a_2} = [0 \quad 0 \quad 0]$

(Z component) $\boldsymbol{b_1} = [1 \quad 1 \quad 0]$ $\quad$ $\boldsymbol{b_2} = [0 \quad 1 \quad 1]$

Let the error operator be $X \otimes I \otimes I \equiv XII = E(\boldsymbol{c}, \boldsymbol{d}) = E([1\ 0\ 0]\,, [0\ 0\ 0])$

Symplectic inner product: $\langle[\boldsymbol{a}, \boldsymbol{b}], [\boldsymbol{c}, \boldsymbol{d}]\rangle_{\text{sym}} \coloneqq \boldsymbol{a}\boldsymbol{d}^T + \boldsymbol{b}\boldsymbol{c}^T$ (modulo 2)

$$\text{Syndrome} = \begin{bmatrix} \langle[\boldsymbol{a_1}, \boldsymbol{b_1}], [\boldsymbol{c}, \boldsymbol{d}]\rangle_{\text{sym}} \\ \langle[\boldsymbol{a}_2, \boldsymbol{b_2}], [\boldsymbol{c}, \boldsymbol{d}]\rangle_{\text{sym}} \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$a_1 d^T + b_1 c^T = [0\ 0\ 0]\ [0\ 0\ 0]^T + [1\ 1\ 0]\ [1\ 0\ 0]^T = 0 + 1 = 1 \text{ (mod 2)}$$

Quantum Parity-Check Matrices

Stabilizers ($n = 3$): $Z \otimes Z \otimes I$ $\quad$ $I \otimes Z \otimes Z$

(X component) $\boldsymbol{a}_1 = [0 \quad 0 \quad 0]$ $\quad$ $\boldsymbol{a}_2 = [0 \quad 0 \quad 0]$

(Z component) $\boldsymbol{b}_1 = [1 \quad 1 \quad 0]$ $\quad$ $\boldsymbol{b}_2 = [0 \quad 1 \quad 1]$

$$H = \left[\begin{array}{ccc|ccc} 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{array}\right] = [H_a \mid H_b] \; ; \; \boldsymbol{H_a H_b^T + H_b H_a^T = 0}$$

Symplectic inner product: $\langle[\boldsymbol{a},\boldsymbol{b}],[\boldsymbol{c},\boldsymbol{d}]\rangle_{\text{sym}} := \boldsymbol{a}\boldsymbol{d}^T + \boldsymbol{b}\boldsymbol{c}^T$ (modulo 2)

$$\text{Syndrome} = \begin{bmatrix} \langle[\boldsymbol{a}_1,\boldsymbol{b}_1],[\boldsymbol{c},\boldsymbol{d}]\rangle_{\text{sym}} \\ \langle[\boldsymbol{a}_2,\boldsymbol{b}_2],[\boldsymbol{c},\boldsymbol{d}]\rangle_{\text{sym}} \end{bmatrix} = H_a d^T + H_b c^T$$

Minimum Distance and Logical Operators

$$H = \left[\begin{array}{ccc|ccc} 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{array}\right] = [H_a \mid H_b] \ ; \ \boldsymbol{H_a H_b^T + H_b H_a^T = 0}$$

$$\text{Syndrome} = \begin{bmatrix} \langle [\boldsymbol{a}_{\boldsymbol{1}}, \boldsymbol{b}_{\boldsymbol{1}}], [\boldsymbol{c}, \boldsymbol{d}] \rangle_{\text{sym}} \\ \langle [\boldsymbol{a}_2, \boldsymbol{b}_{\boldsymbol{2}}], [\boldsymbol{c}, \boldsymbol{d}] \rangle_{\text{sym}} \end{bmatrix} = H_a d^T + H_b c^T$$

What are the “codewords” of this quantum code?

Generated by $\bar{X} = [1\ 1\ 1\ , 0\ 0\ 0]$ and $\bar{Z} = [0\ 0\ 0\ , 1\ 0\ 0]$

Minimum Distance: Minimum weight of any codeword

Codewords are referred to as logical operators

Poll Question 19

Given the parity-check matrix $H = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}$ of the code, what is the binary syndrome for the error XXX?

$A.$ $[0,0]^T$
$B.$ $[0,1]^T$
$C.$ $[1,0]^T$
$D.$ $[1,1]^T$

E. I'm not sure

Stabilizers and Logical Operators

$$H = \left[\begin{array}{ccc|ccc} 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{array}\right] = [H_a \mid H_b] \; ; \; \boldsymbol{H_a H_b^T + H_b H_a^T = 0}$$

Minimum Distance: Minimum weight of any codeword
Codewords are referred to as logical operators

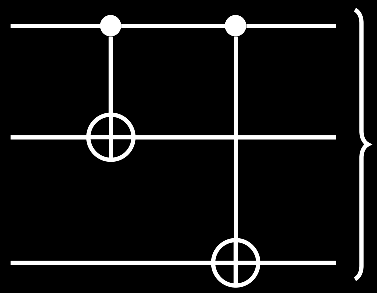
Generated by $\bar{X} = [1\,1\,1\,,0\,0\,0]$ and $\bar{Z} = [0\,0\,0\,,1\,0\,0]$

[[3,1,1]] Code

$|\psi\rangle_L = \alpha\,|0\rangle + \beta\,|1\rangle$

Z $|0\rangle$

$|0\rangle$



$\overline{|\psi\rangle} = \alpha\,|000\rangle + \beta\,|111\rangle$

Stabilizers and Logical Operators

$$H = \left[\begin{array}{ccc|ccc} 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{array}\right] = [H_a \mid H_b] \; ; \; \boldsymbol{H_a H_b^T + H_b H_a^T = 0}$$

Minimum Distance: Minimum weight of any codeword

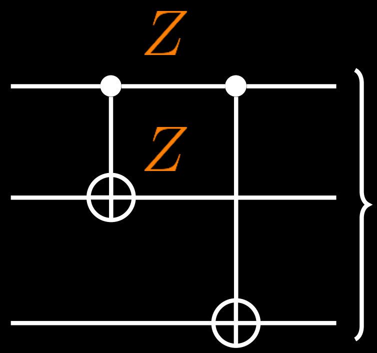
Codewords are referred to as logical operators

Generated by $\bar{X} = [1\,1\,1\,,0\,0\,0]$ and $\bar{Z} = [0\,0\,0\,,1\,0\,0]$

$|\psi\rangle_L = \alpha|0\rangle + \beta|1\rangle$

$|0\rangle$

$|0\rangle$



$\overline{|\psi\rangle} = \alpha|000\rangle + \beta|111\rangle$

[[3,1,1]] Code

Stabilizers and Logical Operators

$$H = \left[\begin{array}{ccc|ccc} 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{array}\right] = [H_a \mid H_b] \; ; \; \boldsymbol{H_a H_b^T + H_b H_a^T = 0}$$

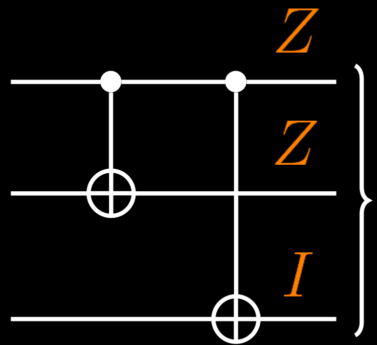
Minimum Distance: Minimum weight of any codeword
Codewords are referred to as logical operators

Generated by $\bar{X} = [1\,1\,1\,,0\,0\,0]$ and $\bar{Z} = [0\,0\,0\,,1\,0\,0]$

$|\psi\rangle_L = \alpha\,|0\rangle + \beta\,|1\rangle$

$|0\rangle$

$|0\rangle$



$\overline{|\psi\rangle} = \alpha\,|000\rangle + \beta\,|111\rangle$

[[3,1,1]] Code

Stabilizers and Logical Operators

$$H = \left[\begin{array}{ccc|ccc} 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{array}\right] = [H_a \mid H_b] \;;\; \boldsymbol{H_a H_b^T + H_b H_a^T = 0}$$

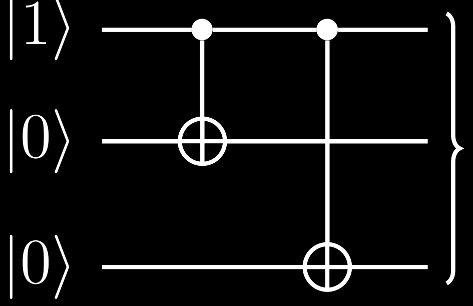
Minimum Distance: Minimum weight of any codeword

Codewords are referred to as logical operators

Generated by $\bar{X} = [1\,1\,1\,,0\,0\,0]$ and $\bar{Z} = [0\,0\,0\,,1\,0\,0]$

[[3,1,1]] Code

$X\,|\psi\rangle_L = \alpha\,|0\rangle + \beta\,|1\rangle$



$\overline{|\psi\rangle} = \alpha\,|000\rangle + \beta\,|111\rangle$

Stabilizers and Logical Operators

$$H = \left[\begin{array}{ccc|ccc} 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{array}\right] = [H_a \mid H_b] \; ; \; \boldsymbol{H_a H_b^T + H_b H_a^T = 0}$$

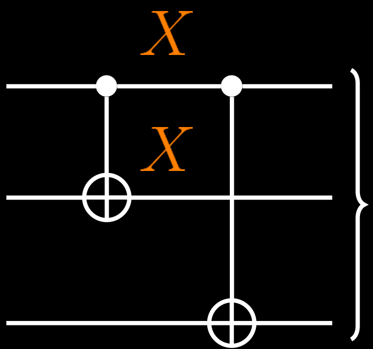
Minimum Distance: Minimum weight of any codeword
Codewords are referred to as logical operators

Generated by $\bar{X} = [1\,1\,1\,,0\,0\,0]$ and $\bar{Z} = [0\,0\,0\,,1\,0\,0]$

$|\psi\rangle_L = \alpha\,|0\rangle + \beta\,|1\rangle$

$|0\rangle$

$|0\rangle$



$\overline{|\psi\rangle} = \alpha\,|000\rangle + \beta\,|111\rangle$

[[3,1,1]] Code

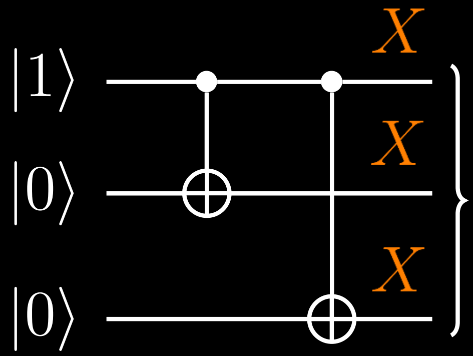
Stabilizers and Logical Operators

$$H = \left[\begin{array}{ccc|ccc} 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{array}\right] = [H_a \mid H_b] \ ; \ \boldsymbol{H_a H_b^T + H_b H_a^T = 0}$$

Minimum Distance: Minimum weight of any codeword
Codewords are referred to as logical operators

Generated by $\bar{X} = [1\ 1\ 1\ , 0\ 0\ 0]$ and $\bar{Z} = [0\ 0\ 0\ , 1\ 0\ 0]$

$$|\psi\rangle_L = \alpha\,|0\rangle + \beta\,|1\rangle$$



$$\overline{|\psi\rangle} = \alpha\,|000\rangle + \beta\,|111\rangle$$

[[3,1,1]] Code

The Steane code

[7,4,3] Hamming Code:

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

[[7,1,3]] Steane Code:

$$H_S = \left[\ H_a \mid H_b\ \right] = \left[\begin{array}{c|c} H & 0 \\ \hline 0 & H \end{array}\right] \begin{array}{c} X \\ Z \end{array}$$

$$H_a H_b^T + H_b H_a^T = \left[\begin{array}{c|c} 0 & HH^T \\ \hline 0 & 0 \end{array}\right] = 0$$

Logical Operators:

$$\bar{X} = \left[\begin{array}{ccccccc|ccccccc} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right]$$

$$\bar{Z} = \left[\begin{array}{ccccccc|ccccccc} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{array}\right]$$

Standard array decoding

ML decoding: $$\hat{x} = \underset{x\in C}{\operatorname{argmin}}\, d(x, y = 110\,101)$$

2^k

2^{n-k}

000 000	001 011	010 110	100 101	011 101	101 110	110 011	111 000	000
000 001	001 010	010 111	100 100	011 100	101 111	110 010	111 001	001
000 010	001 001	010 100	100 111	011 111	101 100	110 001	111 010	010
000 100	001 111	010 010	100 001	011 001	101 010	110 111	111 100	100
001 000	000 011	011 110	101 101	010 101	100 110	111 011	110 000	011
010 000	011 011	000 110	110 101	001 101	111 110	100 011	101 000	110
100 000	101 011	110 110	000 101	111 101	001 110	010 011	011 000	101

Complexity scales exponentially!!

The Steane code: correcting errors

[[7,1,3]] Steane Code:

$$H_S = \left[\begin{array}{c|c} H_a & H_b \end{array}\right] = \begin{array}{c} \left[\begin{array}{c|c} H & 0 \\ \hline 0 & H \end{array}\right] \end{array}\begin{array}{c} X \\ Z \end{array}$$

$$\text{Syndrome} = \begin{bmatrix} \langle[\boldsymbol{a}_{\mathbf{1}}, \boldsymbol{b}_{\mathbf{1}}], [\boldsymbol{c}, \boldsymbol{d}]\rangle_{\text{sym}} \\ \vdots \\ \langle[\boldsymbol{a}_{14}, \boldsymbol{b}_{\mathbf{14}}], [\boldsymbol{c}, \boldsymbol{d}]\rangle_{\text{sym}} \end{bmatrix} = H_a d^T + H_b c^T$$

Syndrome decoding: Given the measured syndrome, determine the most likely error $[c, d]$ that matches the measured syndrome

Complexity scales exponentially!!

Poll Question 20

Given the Steane code parity-check matrix $H_S = [H_a \mid H_b] = \begin{bmatrix} H & 0 \\ 0 & H \end{bmatrix}$ with

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

what is the most likely error pattern corresponding to the syndrome $[0,1,0,0,0,0]^T$? Assume that the channel is memoryless, so it applies independent Pauli errors.

$A.$ $[0\ 0\ 0\ 0\ 0\ 0\ 0 \mid 1\ 0\ 0\ 1\ 0\ 0\ 0]$
$B.$ $[1\ 0\ 0\ 1\ 0\ 0\ 0 \mid 0\ 0\ 0\ 0\ 0\ 0\ 0]$
$C.$ $[0\ 0\ 0\ 0\ 0\ 0\ 0 \mid 0\ 0\ 0\ 0\ 0\ 1\ 1]$
$D.$ $[0\ 0\ 0\ 0\ 0\ 0\ 0 \mid 0\ 1\ 0\ 0\ 0\ 0\ 0]$
$E.$ $[0\ 1\ 0\ 0\ 0\ 0\ 0 \mid 0\ 0\ 0\ 0\ 0\ 0\ 0]$

F. I’m not sure

CSS (Calderbank-Shor-Steane) Codes

- Consider two classical codes C_X and C_Z whose parity-check matrices H_X and H_Z satisfy $H_X H_Z^T = 0$

- Define the CSS (stabilizer) code by $H_{\mathrm{CSS}} = \begin{bmatrix} H_X & 0 \\ 0 & H_Z \end{bmatrix}$

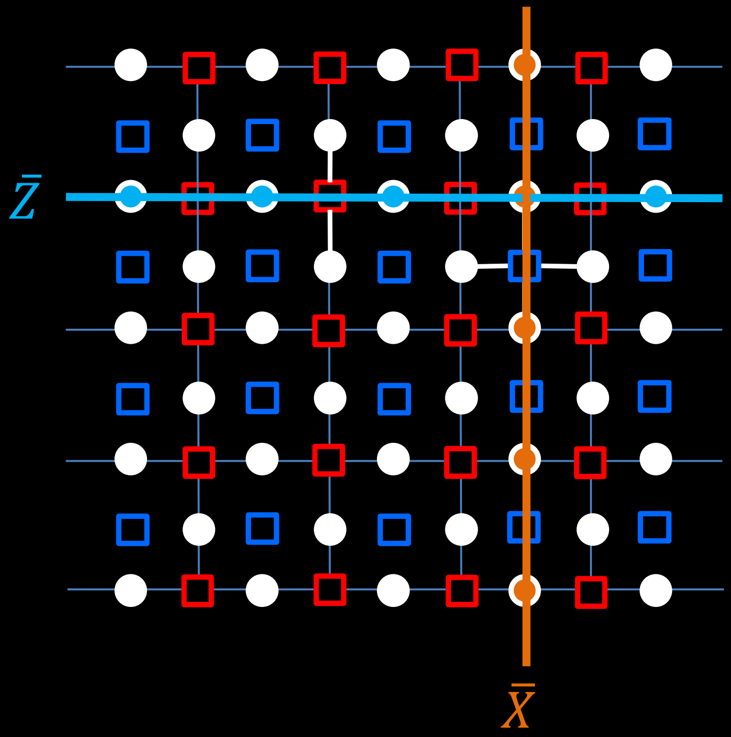
- Logical operators $[c, d]$ defined by $H_X d^T + H_Z c^T = 0$

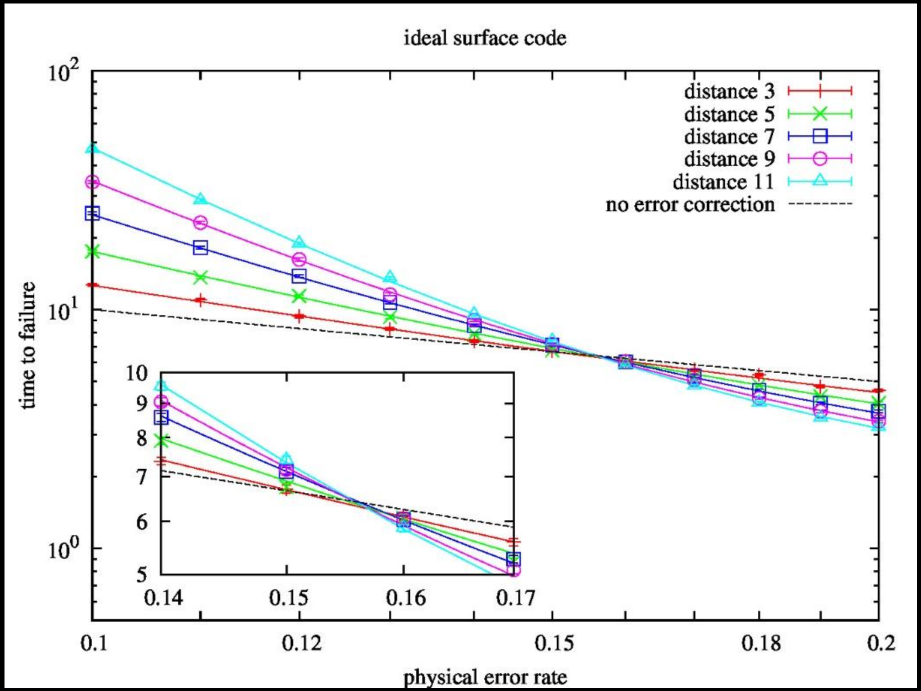
- Error $[e_X, e_Z] \Rightarrow$ syndrome is $s = H_X e_Z^T + H_Z e_X^T$ (mod 2)

- $[[\, n, k, d \,]] = [[\, n, k_X + k_Z - n, w_{\min}([C_X \backslash \mathrm{C}_Z^{\perp}] \cup [C_Z \backslash \mathrm{C}_X^{\perp}]) \,]]$

Surface code

$[[\,O(L^2), 1, L\,]]$





□ - vertex checks (H_X)

□ - plaquette checks (H_Z)

● - Logical Z

● - Logical X

Wang et al. http://arxiv.org/abs/0905.0531

Poll Question 21

Consider the following statements and answer if they are true or false:

1. Errors that produce a zero syndrome must be stabilizers or logical operators
2. The surface code stabilizer generators each involve either 3 or 4 qubits

A. 1 is True, 2 is False

B. 1 is False, 2 is True

C. 1 is True, 2 is True

D. 1 is False, 2 is False

E. I’m not sure

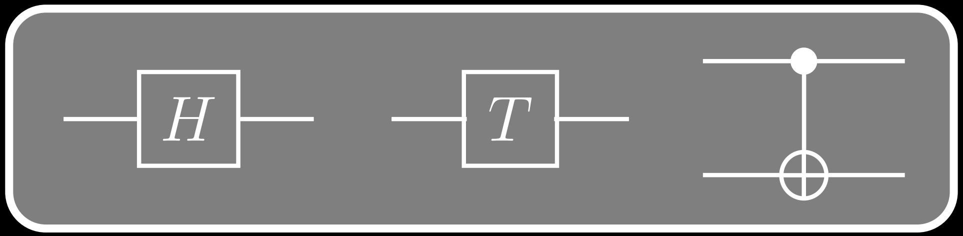
Manipulating encoded information

$$\begin{tikzcd}[column sep=huge, row sep=huge]
k \text{ qubits} & |\psi\rangle_L \arrow[rr, "\substack{\text{Noisy}\\ \text{Logical operation}}"] \arrow[d, "\substack{\text{QECC}\\ \text{Encode}}"'] & & |\phi\rangle_L \\
n \text{ qubits} & \overline{|\psi\rangle} \arrow[rr, "\substack{\text{Physical operation}\\ \text{Fault-tolerant}}"'] \arrow[r, phantom, shift left=6ex, "\substack{\text{Translate}\\ \text{(Synthesize)}}"] & {} & \overline{|\phi\rangle} \arrow[u, "\substack{\text{QECC}\\ \text{Decode}}"']
\end{tikzcd}$$

QECC: Quantum Error Correcting Code

Universal fault-tolerance

Universal gates on k qubits:



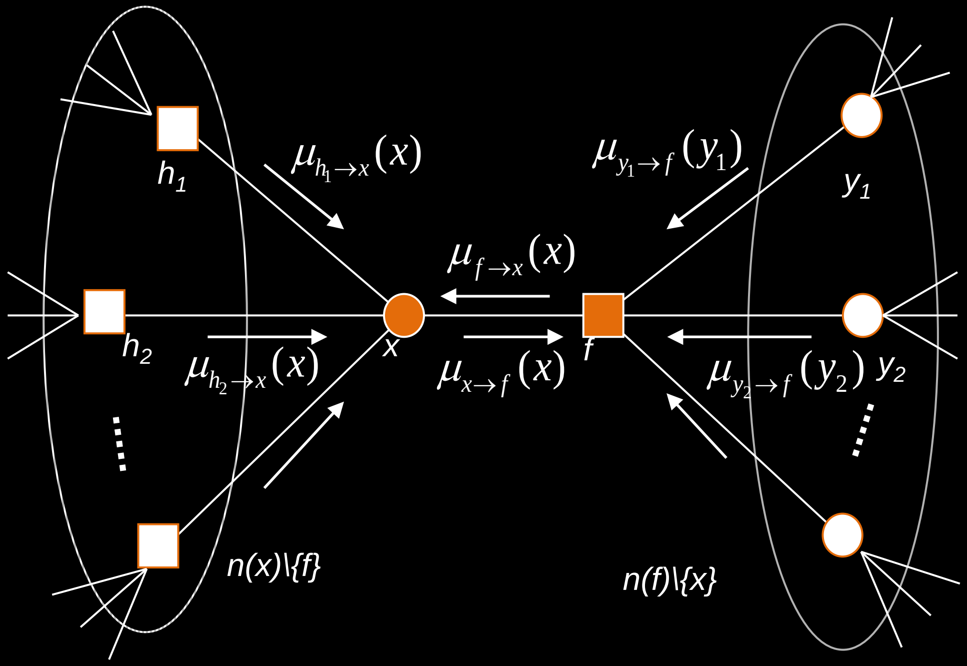
$$\begin{tikzcd}
k \text{ qubits} & |\psi\rangle_L \arrow[r, "\substack{\text{Noisy}\\ \text{Logical operation}}"] \arrow[d, "\substack{\text{QECC}\\ \text{Encode}}"'] \arrow[dr, phantom, "\substack{\text{Translate}\\ \text{(Synthesize)}}"] & |\phi\rangle_L \\
n \text{ qubits} & \overline{|\psi\rangle} \arrow[r, "\substack{\text{Physical operation}\\ \text{Fault-tolerant}}"'] & \overline{|\phi\rangle} \arrow[u, "\substack{\text{QECC}\\ \text{Decode}}"']
\end{tikzcd}$$

Quantum LDPC codes

- Consider two **classical LDPC codes** C_X and C_Z whose parity-check matrices H_X and H_Z satisfy $H_X H_Z^T = 0$
- Define the CSS QLDPC code by $H_{\mathrm{QLDPC}} = \begin{bmatrix} H_X & 0 \\ 0 & H_Z \end{bmatrix}$
- Several QLDPC code families exist:
  - Hypergraph Product codes, e.g., the surface code
  - Bicycle and Generalized Bicycle codes
  - Homological Product codes
  - Lifted Product codes
  - Quantum Tanner codes

Syndrome-based iterative decoding

Belief propagation (BP)



Variable node (VN) update:

$$\mu_{x\to f}(x) = \prod_{h\in n(x)\backslash\{f\}} \mu_{h\to x}(x)$$

Check node (CN) update:

$$\mu_{f\to x}(x) = \sum_{\sim\{x\}}\left(f(X) \prod_{h\in n(f)\backslash\{x\}} \mu_{y\to f}(y)\right)$$

Variable node (VN) decision:

$$g_i(x_i) = \prod_{h\in n(x_i)} \mu_{h\to x_i}(x_i)$$

Poll Question 22

Consider a CSS QLDPC code constructed from classical codes C_X and C_Z. Then which of the following is false?

A. H_X and H_Z are orthogonal
B. H_X and H_Z are sparse, i.e., have very few 1s
C. The code is a stabilizer code
D. Any stabilizer code is a CSS code
E. Universal computation requires fault-tolerant realizations of $H, T, CNOT$ on the logical qubits

F. I’m not sure

Error Correction: Classical vs Quantum

- Classical: Decode based on received vector
  Quantum: Decode based only on measured syndrome

- Classical: Any sparse parity-check matrix gives LDPC
  Quantum: Need two sparse matrices that are orthogonal

- Classical: Only the zero vector causes trivial syndrome
  Quantum: All stabilizers have zero syndrome (degeneracy)

- Classical: Hardware noise quite low, mainly channel noise
  Quantum: Everything noisy – decoding + logical gates

Challenges in QEC

- How to fully leverage degeneracy in QLDPC decoders?
- Local iterative algorithms that correct many errors?
- Can we physically realize good QLDPC codes in hardware despite their many long-range connections?
- Universal fault-tolerance on good QLDPC codes?
- … and many more!

Center for
Quantum Networks
NSF Engineering Research Center

Course Evaluation Survey

We value your feedback on all aspects of this short course. Please go to the link provided in the Zoom Chat or in the email you will soon receive to give your opinions of what worked and what could be improved.

CQN Winter School on Quantum Networks

Funded by National Science Foundation Grant #1941583