Lecture 13: Quantum Error-Correction

Codes, Stabilizer Formalism, and Fault-Tolerance

Moody T. Chu

North Carolina State University

Spring 2026

Lecture Overview

1. Introduction to Quantum Error-Correction
2. Fundamental Quantum Codes
3. General Theory of Error-Correction
4. Classical Linear and CSS Codes
5. The Stabilizer Formalism
6. Fault-Tolerant Quantum Computation

The Problem of Noise

- Noise is a great bane of information processing systems.
- While modern classical computer components are extremely reliable (failure rate below 10^{-17}), many systems still require error-correcting codes to protect against noise.
- The key idea is to encode messages by adding redundant information.
- If noise corrupts the encoded message, the redundancy allows us to decode and recover the original information.

Classical Error-Correction: Repetition Code

- Consider a binary symmetric channel where a bit flips with probability $p > 0$ and is untouched with probability $1 - p$.
- We can protect a bit by replacing it with three copies: $0 \to 000$ and $1 \to 111$.
- At the receiver's end, we decode using majority voting.
- The probability of error drops from p to $p_e = 3p^2 - 2p^3$.
  - Two bit flips $\Rightarrow 3p^2(1-p)$.
  - Three bit flips $\Rightarrow p^3$
  - Total error probability $p_e = 3p^2 - 2p^3$.
- If $p < 1/2$, then $p - p_e = p(1-p)(1-2p) > 0$.
- This repetition code adds redundancy depending on the channel's noise severity.

Challenges for Quantum Error-Correction

- Protecting quantum states faces three formidable difficulties:
  - ◇ No cloning: We cannot simply duplicate a quantum state three times due to the no-cloning theorem.
  - ◇ Errors are continuous: A continuum of different errors can occur on a single qubit, seemingly requiring infinite precision to correct.
  - ◇ Measurement destroys quantum information: Observing the output to decide on a decoding procedure generally destroys the quantum state.
- These problems can be addressed.

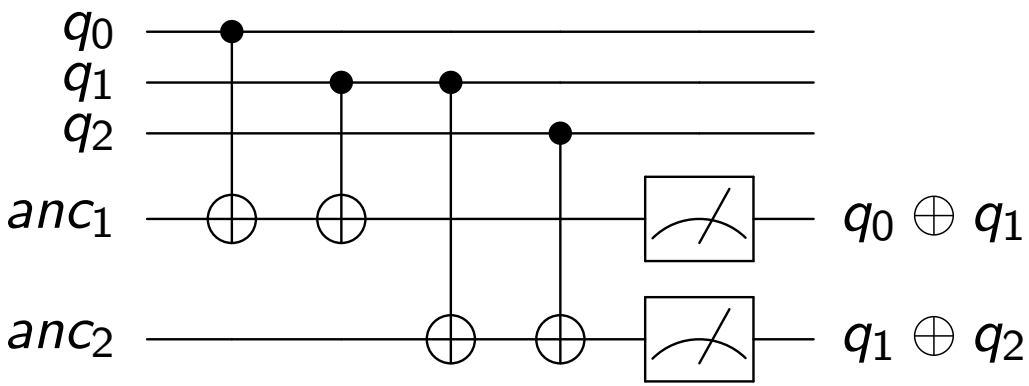
The Three-Qubit Bit Flip Code

- The bit flip channel leaves a qubit untouched with probability $1 - p$ and applies the Pauli X operator with probability p.
- We encode a single qubit state $a|0\rangle + b|1\rangle$ into three qubits as $a|000\rangle + b|111\rangle$.
  - The message is contained in (a, b).
- The logical states are defined as $|0_L\rangle \equiv |000\rangle$ and $|1_L\rangle \equiv |111\rangle$.
  - This encoding is performed via a simple circuit using two CNOT gates.
  - However, we cannot look at the data directly, since any measurement causes the quantum state to collapse, and the information (a and b) is destroyed.

Bit Flip Code: Error Detection & Recovery

- **Syndrome diagnosis** is a clever way to determine the error syndrome without perturbing the amplitudes a and b.
- Equivalent to measuring observables Z_1Z_2 and Z_2Z_3 to compare adjacent qubits without destroying the superposition.
  - ◇ The idea is to use the Z gate to flip signs.
  - ◇ In actual implementation, use $CNOT$ gates.
- Recovery: Based on the syndrome diagnosis, apply the corresponding bit flip (e.g., X_1 if syndrome is 1) to recover the exact initial state.

Bit Flip Code: The Circuit



- q_0, q_1, q_2: Data qubits maintaining the logical superposition.
- anc_1, anc_2: Ancilla qubits initialized to $|0\rangle$ to extract syndrome info.
- CNOT gates: Copy the parity (relationship) of data qubits onto ancillas.
- Meter: Reads the ancillas to generate the error syndrome:
  - ◇ 0: The measured pair matches (No error).
  - ◇ 1: The measured pair differs (Error detected).

Syndrome Table

q_0	q_1	q_2	Error	anc_1	anc_2	Diagnosis
0	0	0	none	0	0	No error
1	0	0	q_0 flipped	1	0	Flip q_0
0	1	0	q_1 flipped	1	1	Flip q_1
0	0	1	q_2 flipped	0	1	Flip q_2

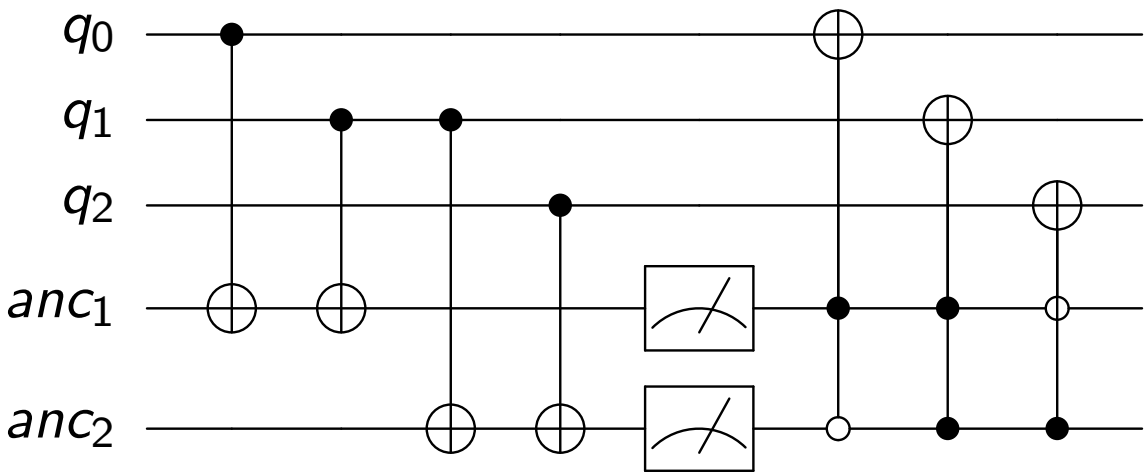
- Each of the four syndromes $(anc_1, anc_2) \in \{00, 10, 11, 01\}$ uniquely identifies which qubit (if any) was flipped, so the error is correctable.

Bit Flip Code: Syndrome Diagnosis

- Key logic:
  - ◇ Syndrome Extraction: CNOTs copy the parity (relative mismatch) onto Ancillas.
  - ◇ No Measurement: Data qubits (q_i) are never observed, preserving superposition.
  - ◇ Diagnosis: Ancilla results identify the bit-flip location.
- Syndrome Table:

State	s_1	s_2	Error
$\|000\rangle$	0	0	None
$\|100\rangle$	1	0	q_0
$\|010\rangle$	1	1	q_1
$\|001\rangle$	0	1	q_2

Bit Flip Code: Syndrome Extraction and Recovery



- In a complete quantum error correction process, the timeline always follows four distinct stages:

  Encoding $\longrightarrow$ Noise (Channel) $\longrightarrow$ Syndrome Extraction $\longrightarrow$ Recovery.

  In the circuit diagram above, only Stage 3 (Syndrome Extraction) and Stage 4 (Recovery) are shown.
- The noise (the unwanted bit-flip X error) occurs right before Column 1 of this diagram, i.e., after the data qubits have been encoded, but before the syndrome extraction starts.

The Three-Qubit Phase Flip Code: The Idea

- The phase flip code is not a different way of detecting errors.
  - ◇ It is essentially the bit flip code translated into a different "language."
- **The key algebraic fact:** In the Hadamard basis $\{|+\rangle, |-\rangle\}$, the Pauli Z gate acts exactly as a *bit flip*:
  - ◇ Recall $|+\rangle = \frac{|0\rangle+|1\rangle}{\sqrt{2}}$ and $|-\rangle = \frac{|0\rangle-|1\rangle}{\sqrt{2}}$. Then:

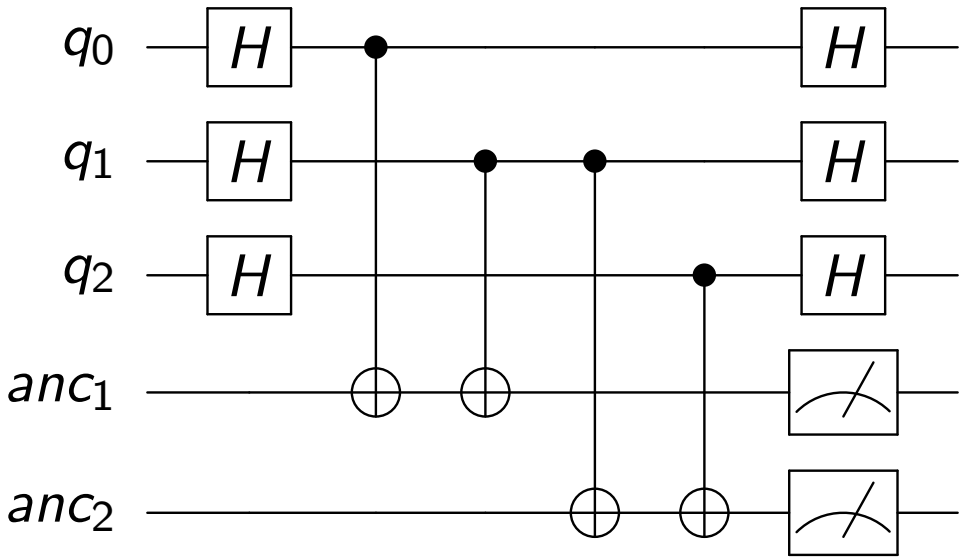
$$Z|+\rangle = \frac{Z|0\rangle + Z|1\rangle}{\sqrt{2}} = \frac{|0\rangle - |1\rangle}{\sqrt{2}} = |-\rangle, \quad Z|-\rangle = |+\rangle.$$

  - ◇ A phase flip (Z) in the computational basis $\{|0\rangle, |1\rangle\}$ becomes a bit flip (X) in the Hadamard basis $\{|+\rangle, |-\rangle\}$.
- In quantum mechanics, if there is a way to fix bit flips ($0 \leftrightarrow 1$), then any error can be fixed by simply translating that error into a bit flip.
  - ◇ That is exactly what the phase flip code does: apply H to rotate into the $\{|+\rangle, |-\rangle\}$ basis, run the bit-flip code, then rotate back.

The Three-Qubit Phase Flip Code: The Setup

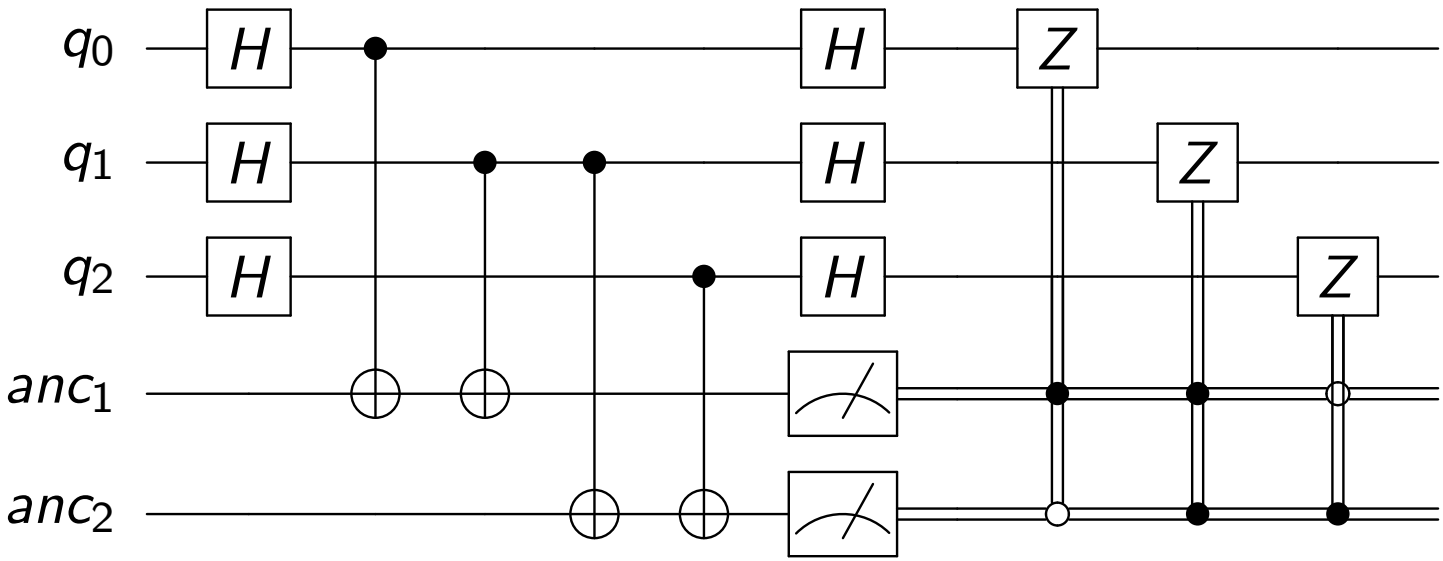
- The phase flip channel applies the Pauli Z operator with probability p, taking $a|0\rangle + b|1\rangle$ to $a|0\rangle - b|1\rangle$.
- In the $|+\rangle$, $|-\rangle$ basis, Z acts as a bit flip.
- Encode the logical states as $|0_L\rangle \equiv |+++\rangle$ and $|1_L\rangle \equiv |---\rangle$.
- The H gate translate the states to the $0, 1$ basis to determine the syndrome diagnosis.
- Error-detection and recovery are identical to the bit flip code but conjugated by Hadamard gates (e.g., measuring X_1X_2 and X_2X_3).

Phase Flip Code: Circuit



- Similar to the three-qubit bit flip circuit, except for the $H^{\otimes 3}$ applications.

Phase Flip Code: Circuit with Recovery



- Solid dot (•) = correction fires when that ancilla measures 1; open dot (◦) = fires when it measures 0.
- Column 7: Z on q_0 iff (anc_1=1, anc_2=0). Column 8: Z on q_1 iff (1,1). Column 9: Z on q_2 iff (0,1).

The Shor Nine-Qubit Code: Why Is This Significant?

- The Shor code is the "gold standard" for simple error correction because it is the first code that achieves universal fault tolerance for single-qubit errors.
  - Single-error limitation: The bit flip code cannot detect phase flips; the phase flip code cannot detect Bit flips.
  - The Y problem: Quantum errors are not just bit or phase flips. A general error is a complex linear combination of I, X, Y, and Z.
  - Universal protection: By correcting both X (Bit) and Z (Phase) errors, the Shor code inherently corrects Y errors (since $Y \propto XZ$).
    - It is the smallest code capable of correcting an arbitrary error on a single qubit.

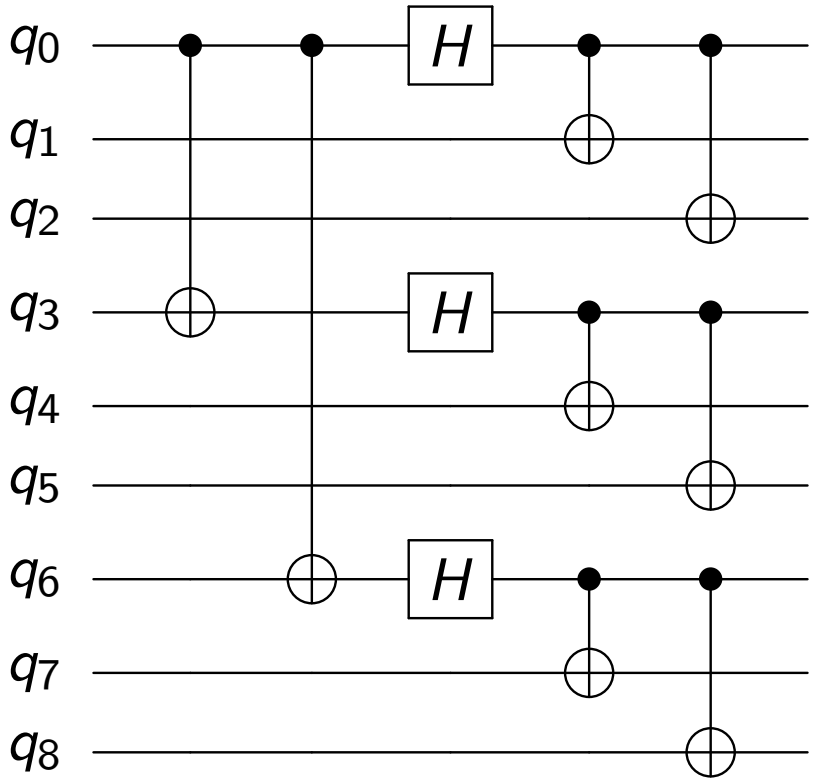
Shor Code: The Setup

- The Shor code combines the phase flip and bit flip codes to protect against an arbitrary error on a single qubit.
- Encoding uses concatenation: first encode using the phase flip code, then encode each resulting qubit using the bit flip code.
  - Logical 0:
$$|0_L\rangle = \frac{1}{\sqrt{8}}(|000\rangle + |111\rangle)^{\otimes 3}$$
  - Logical 1:
$$|1_L\rangle = \frac{1}{\sqrt{8}}(|000\rangle - |111\rangle)^{\otimes 3}$$
  - Any valid message is simply a combination, $\alpha|0_L\rangle + \beta|1_L\rangle$.
- It corrects bit flips by comparing qubits within (inner) blocks, and phase flips by comparing the (outer) signs between blocks.

Shor Code: Correcting Arbitrary Errors

- The Shor code protects against completely arbitrary errors on a single qubit.
- Any error $\mathcal{E}$ can be expanded as a linear combination of I, X_1, Z_1, and $X_1 Z_1$.
- Measuring the error syndrome collapses the continuous superposition into one of the four discrete error states, from which we can recover.
- By correcting a discrete set of errors, a quantum code automatically corrects a continuous class of errors.

Shor Code: Encoding Circuit



- Columns 1 & 2: Phase-flip encoding leaders ($\text{CNOT}_{0\to3}$ and $\text{CNOT}_{0\to6}$).
- Column 3: Basis rotation (H on leaders q_0, q_3, q_6).
- Columns 4 & 5: Inner bit-flip encoding within each 3-qubit block.

Shor Code: Tracing Column 1 & 2

- Initial input state:

$$|\Psi_0\rangle = \alpha|000\rangle|000\rangle|000\rangle + \beta|100\rangle|000\rangle|000\rangle$$

  - ◇ The message starts on $q_0 = \alpha\,|0\rangle + \beta\,|1\rangle$, with all 8 ancillas at $|0\rangle$.
  - ◇ Group them into three 3-qubit blocks.
- Columns 1 & 2: Macroscopic Copying.
  - ◇ Gates: CNOT($q_0 \to q_3$) and CNOT($q_0 \to q_6$).
  - ◇ Mechanism: The block leaders (q_0, q_3, q_6) copy the computational basis state of q_0.

$$|\Psi_2\rangle = \alpha|000\rangle|000\rangle|000\rangle + \beta|100\rangle|100\rangle|100\rangle$$

Shor Code: Tracing Column 3

- Column 3: Basis Translation.
  - ◇ Gates: Hadamards (H) on the block leaders: q_0, q_3, q_6.
  - ◇ Mechanism: Rotates the block leaders into the phase basis, establishing the outer phase-flip protection code.
- State after column 3 ($|\Psi_3\rangle$).
  - ◇ Substituting $H|0\rangle = |+\rangle$ and $H|1\rangle = |-\rangle$.
  - ◇

$$|\Psi_3\rangle = \frac{\alpha}{\sqrt{8}}(|000\rangle + |100\rangle)^{\otimes 3} + \frac{\beta}{\sqrt{8}}(|000\rangle - |100\rangle)^{\otimes 3}$$

Shor Code: Tracing Columns 4 & 5

- Columns 4 & 5: Inner Bit-Flip Protection.
  - ◇ Gates: Internal CNOT pairs within each block:
    - ○ Block 1: $q_0 \to q_1$, then $q_0 \to q_2$
    - ○ Block 2: $q_3 \to q_4$, then $q_3 \to q_5$
    - ○ Block 3: $q_6 \to q_7$, then $q_6 \to q_8$
  - ◇ Mechanism: If a block leader is $|1\rangle$, it flips its targets, transforming internal $|100\rangle$ terms into $|111\rangle$.
- Final Encoded Logical State ($|\Psi_{\text{final}}\rangle$). (Show this!)

$$|\Psi_{\text{final}}\rangle = \alpha|0_L\rangle + \beta|1_L\rangle$$

Subspace Embedding & Logical State Encoding

- An unencoded physical qubit lives in a fragile 2-dimensional Hilbert space ($\mathbb{C}^2$). The Shor code maps this state into a massive 512-dimensional space ($\mathbb{C}^{512}$).
- We restrict our logical state to a specific 2-dimensional code subspace $\mathcal{C} \subset \mathbb{C}^{512}$ spanned by:

$$\begin{aligned} |0_L\rangle &= \frac{1}{\sqrt{8}}(|000\rangle + |111\rangle)^{\otimes 3}, \\ |1_L\rangle &= \frac{1}{\sqrt{8}}(|000\rangle - |111\rangle)^{\otimes 3}. \end{aligned}$$

- The arbitrary logical message $|\Psi_L\rangle = \alpha\,|0_L\rangle + \beta\,|1_L\rangle$ preserves the exact information content of the input, but embeds it non-locally.
- Let $|\phi_\pm\rangle := \frac{1}{\sqrt{2}}(|000\rangle \pm |111\rangle)$. Write the uncorrupted message as:

$$|\Psi_L\rangle = \alpha\,|\phi_+\rangle\,|\phi_+\rangle\,|\phi_+\rangle + \beta\,|\phi_-\rangle\,|\phi_-\rangle\,|\phi_-\rangle$$

Noise Impact & Superposition Protection

- Suppose a local noise strikes, causing a bit-flip (X_0) on the first physical qubit. Then, the state is kicked out of the code subspace $\mathcal{C}$ into a disjoint, orthogonal error subspace $\mathcal{E}_{X_0}$.

$$X_0 |\Psi_L\rangle = \frac{\alpha}{\sqrt{2}}(|100\rangle + |011\rangle) |\phi_+\rangle^{\otimes 2} + \frac{\beta}{\sqrt{2}}(|100\rangle - |011\rangle) |\phi_-\rangle^{\otimes 2}$$

$$= \frac{1}{\sqrt{2}} |100\rangle \otimes \left(\alpha |\phi_+\rangle^{\otimes 2} + \beta |\phi_-\rangle^{\otimes 2}\right) + \frac{1}{\sqrt{2}} |011\rangle \otimes \left(\alpha |\phi_+\rangle^{\otimes 2} - \beta |\phi_-\rangle^{\otimes 2}\right)$$

$$= \alpha \left|0_L^{(X_0)}\right\rangle + \beta \left|1_L^{(X_0)}\right\rangle \quad (\in \mathcal{E}_{X_0})$$

where the basis vectors of $\mathcal{E}_{X_0}$ are defined as:

$$\left|0_L^{(X_0)}\right\rangle = \frac{|100\rangle + |011\rangle}{\sqrt{2}} |\phi_+\rangle^{\otimes 2}, \quad \left|1_L^{(X_0)}\right\rangle = \frac{|100\rangle - |011\rangle}{\sqrt{2}} |\phi_-\rangle^{\otimes 2}$$

- **No Data Leakage:** Because $\mathcal{C} \perp \mathcal{E}_{X_0}$, detecting the error gives the exact same result regardless of α and β, preventing state collapse.
- **Container vs. Content:** We measure *where* the state landed (the error subspace), not *what* state it is, leaving the quantum superposition intact.

Measuring Subspaces: Stabilizer Operators

- **What is a Stabilizer? (The "Container Check")** A stabilizer S is an operator that leaves every state in the code subspace $\mathcal{C}$ invariant:

$$S|\Psi_L\rangle = +1|\Psi_L\rangle \quad \forall |\Psi_L\rangle \in \mathcal{C}$$

  Because S yields $+1$ deterministically, measuring S extracts **zero information** about α or β.

- **How Stabilizers Flag Errors (Commutation Rules)** If noise E strikes, measuring S returns the eigenvalue ± 1:

$$S(E|\Psi_L\rangle) = \begin{cases} +1(E|\Psi_L\rangle) & \text{if } SE = ES \quad (\text{State is in } \mathcal{C}) \\ -1(E|\Psi_L\rangle) & \text{if } SE = -ES \quad (\text{State shifted to } \mathcal{E}_E) \end{cases}$$

- **The 8 Stabilizers of the Shor Code**:
  - **Bit-flip (Z-type):** $Z_0Z_1, Z_1Z_2, Z_3Z_4, Z_4Z_5, Z_6Z_7, Z_7Z_8$
    (An X_0 error anti-commutes with Z_0Z_1, flipping its eigenvalue to -1).
  - **Phase-flip (X-type):** $X_0X_1X_2X_3X_4X_5$ and $X_3X_4X_5X_6X_7X_8$
    (Compares phase signs between the three 3-qubit blocks).

Full Error Coverage: Handling X, Y, and Z Noise

- **Error Discretization (Pauli Basis)**
  Any arbitrary single-qubit noise $E = c_I I + c_X X + c_Y Y + c_Z Z$ collapses into a discrete Pauli error (I, X, Y, or Z) upon syndrome measurement.
- **Bit-Flip Syndromes (X Errors — 6 Ancillas)**
  Measures intra-block parities ($Z_i Z_j$) *within* each 3-qubit block:

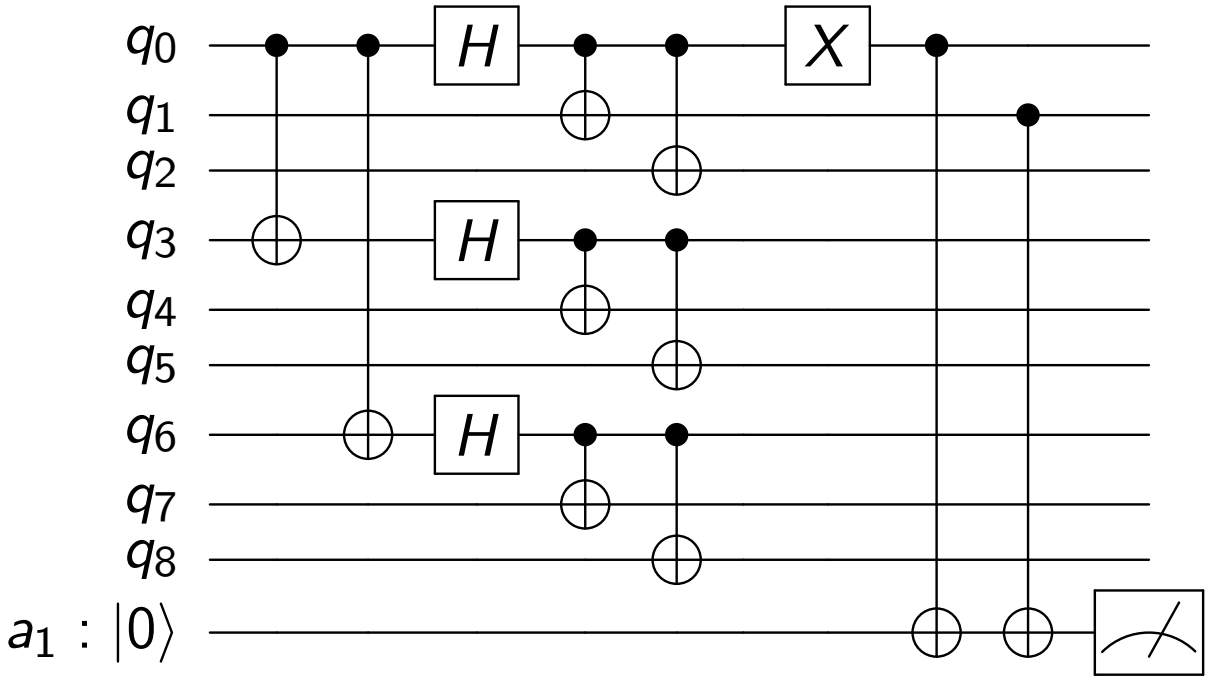
$$(q_0 \oplus q_1,\ q_1 \oplus q_2), \quad (q_3 \oplus q_4,\ q_4 \oplus q_5), \quad (q_6 \oplus q_7,\ q_7 \oplus q_8)$$

- **Phase-Flip Syndromes (Z Errors — 2 Ancillas)**
  Measures inter-block phase parities *between* the three 3-qubit blocks:

$$\text{Block 1 vs. Block 2} \quad \text{and} \quad \text{Block 2 vs. Block 3}$$

- **Combined Errors ($Y = iXZ$)**
  A Y error acts as both a bit flip and a phase flip simultaneously, triggering **both** an intra-block X syndrome and an inter-block Z syndrome.

Circuit Walkthrough: Encoding, Error, & Syndrome



- Columns 1-5 (Encoding): The Shor circuit distributes the state across the 9 qubits.
- Column 7 (Noise): An environmental bit-flip error (X) strikes q_0.
- Columns 8-10 (Syndrome): Ancilla a_1 safely extracts the Z_0Z_1 parity using CNOTs. Measuring $a_1 = 1$ confirms the mismatch without leaking data or collapsing the coefficients α and β.

State Recovery & Subspace Restitution

- Once the syndrome measurement returns a set of ± 1 eigenvalues, we map them directly to a unique error configuration.
  - ◇ De-multiplexing the syndrome: The classical computer reads the syndrome. If $Z_0 Z_1 = -1$ and $Z_1 Z_2 = +1$, it diagnoses a bit-flip on q_0.
  - ◇ Targeted Operator Application: We apply the inverse operator (which for Pauli gates is the gate itself) to the designated qubit:

$$X_0 \left|\Psi_{\text{error}}\right\rangle = X_0(X_0 \left|\Psi_L\right\rangle) = \left|\Psi_L\right\rangle$$

  - ◇ Subspace Restitution: This operation rotates the state cleanly back into the 2D code subspace $\mathcal{C}$.

General Theory of Quantum Error-Correction

- A quantum error-correction code is a subspace C of a larger Hilbert space, with projector P.
- Noise is described by a quantum operation $\mathcal{E}$, and recovery by a trace-preserving operation $\mathcal{R}$.
- Successful error-correction requires

$$(\mathcal{R} \circ \mathcal{E})(\rho) \propto \rho$$

  for any state ρ with support in the code C.
- The subspaces corresponding to different error syndromes must be orthogonal and *isometric* copies of the original code space.

$$PF_i^\dagger F_j P = c_i \delta_{ij} P$$

  - ◇ Orthogonal subspaces: error subspaces $F_i C$ and $F_j C$ are distinct and non-overlapping, enabling reliable syndrome measurement.
  - ◇ Isometric (undeformed): each error F_i maps orthogonal codewords to orthogonal states, preserving inner products within the code.

QEC Conditions (Theorem 10.1)

Theorem (Knill–Laflamme, 1997)

Let $\mathcal{C}$ be a quantum code and P be the projector onto $\mathcal{C}$. Suppose $\mathcal{E}$ is a quantum operation with operation elements $\{E_i\}$. A necessary and sufficient condition for the existence of an error-correction operation $\mathcal{R}$ correcting $\mathcal{E}$ on $\mathcal{C}$ is:

$$PE_i^\dagger E_j P = \alpha_{ij} P$$

for some Hermitian, positive semi-definite matrix $\alpha = [\alpha_{ij}]$.

- α is automatically positive semi-definite: the necessity proof shows $\alpha_{ij} = \sum_k c_{ki}^* c_{kj}$, which is a Gram matrix.
- If this condition holds, $\{E_i\}$ constitutes a correctable set of errors.
- **Proof strategy:**
  - **Sufficiency:** Given the equation, explicitly construct a recovery $\mathcal{R}$.
  - **Necessity:** Given that a perfect $\mathcal{R}$ exists, derive the equation.

Proof of Sufficiency: Step 1 (Diagonalization)

- **Assumption:** $PE_i^\dagger E_j P = \alpha_{ij} P$.
- Diagonalize α: $d = u^\dagger \alpha u$, where u is unitary and d is diagonal.
- Define a new, equivalent set of operation elements for the noise $\mathcal{E}$:

$$F_k := \sum_i u_{ik} E_i.$$

- Verify that $\{F_k\}$ satisfies a diagonalized QEC condition. Using the definition of F_k:

$$PF_k^\dagger F_l P = \sum_{i,j} u_{ik}^* u_{jl}\, PE_i^\dagger E_j P = \sum_{i,j} u_{ik}^* u_{jl}\, \alpha_{ij} P = (u^\dagger \alpha u)_{kl} P = d_{kl} P.$$

  - ◇ Since d is diagonal, $d_{kl} = 0$ for $k \neq l$. We have reshaped the errors into a set $\{F_k\}$ that acts orthogonally on the code space.

Proof of Sufficiency: Step 2 (Polar Decomposition)

- The polar decomposition of an operator A states that $A = U|A|$, where U is a partial isometry (or unitary on the subspace) and $|A| = \sqrt{A^\dagger A}$ is a positive operator.
- Applying this to $F_k P$:

$$F_k P = U_k \sqrt{P F_k^\dagger F_k P} = U_k \sqrt{d_{kk} P} = \sqrt{d_{kk}} U_k P$$

  for some unitary U_k.
- Physical meaning: The error F_k acting on the code space does two things:
  - ◇ It shrinks the state by a constant factor $\sqrt{d_{kk}}$.
  - ◇ It rotates the state into a new subspace via the unitary U_k.

Proof of Sufficiency: Step 3 (Orthogonal Subspaces)

- Define projectors for the error subspaces: $P_k \equiv U_k P U_k^\dagger$.
- From $F_k P = \sqrt{d_{kk}}\, U_k P$ (polar decomp.): $P U_k^\dagger = \frac{1}{\sqrt{d_{kk}}} P F_k^\dagger$. Use this to check mutual orthogonality:

$$P_l P_k = U_l P U_l^\dagger U_k P U_k^\dagger = \frac{1}{\sqrt{d_{ll} d_{kk}}} U_l P (F_l^\dagger F_k) P U_k^\dagger$$

$$= \frac{d_{kl}}{\sqrt{d_{ll} d_{kk}}} U_l P U_k^\dagger = 0 \quad (k \neq l, \text{ since } d_{kl} = 0).$$

- The errors F_k rotate the code into mutually orthogonal, non-overlapping subspaces P_k — each error is detectable without disturbing the logical data.

Proof of Sufficiency: Step 4 (The Recovery Operation)

- Construct the recovery operation $\mathcal{R}$ using the projectors P_k and the reversing unitaries $U_k^\dagger$:

$$\mathcal{R}(\sigma) := \sum_k U_k^\dagger P_k \sigma P_k U_k$$

- Test this on an error-corrupted state $\mathcal{E}(\rho) = \sum_l F_l \rho F_l^\dagger$:

$$\begin{aligned}\mathcal{R}(\mathcal{E}(\rho)) &= \sum_{k,l} U_k^\dagger P_k F_l \rho F_l^\dagger P_k U_k \\ &= \sum_{k,l} \delta_{kl} d_{kk} \rho \quad (\text{using } \rho = P\rho P, \quad P_k F_l P = \delta_{kl} F_k P) \\ &\propto \rho\end{aligned}$$

- The original state ρ is perfectly restored! This completes the proof of sufficiency.

Proof of Necessity: Step 1

- Suppose an error-correction operation $\mathcal{R}$ with elements $\{R_j\}$ perfectly corrects the noise $\mathcal{E}$ on the code space P.
- Mathematically, for any state ρ, the combined process acting on the encoded state $P\rho P$ must return the state proportional to itself:

$$\mathcal{R}(\mathcal{E}(P\rho P)) = \sum_{i,j} R_j E_i P \rho P E_i^\dagger R_j^\dagger = cP\rho P$$

  where c is a constant.
- Because this holds for *all* ρ, the operation elements on the left must be proportional to the operation element on the right (P). Thus:

$$R_k E_i P = c_{ki} P$$

  for some complex numbers c_{ki}. (Prove this!)

Proof of Necessity: Step 2

- We have the relation:

$$R_k E_i P = c_{ki} P \quad \Longrightarrow \quad P E_i^\dagger R_k^\dagger = c_{ki}^* P$$

- Multiplying these two expressions together for indices i and j:

$$P E_i^\dagger R_k^\dagger R_k E_j P = c_{ki}^* c_{kj} P$$

- Because $\mathcal{R}$ is a valid, trace-preserving quantum operation, its elements must satisfy the completeness relation: $\sum_k R_k^\dagger R_k = I$.

- Summing our equation over all k:

$$P E_i^\dagger \left(\sum_k R_k^\dagger R_k \right) E_j P = \left(\sum_k c_{ki}^* c_{kj} \right) P$$

$$P E_i^\dagger E_j P = \alpha_{ij} P$$

where $\alpha_{ij} \equiv \sum_k c_{ki}^* c_{kj}$ is a Hermitian matrix. This exactly matches our theorem condition, completing the proof!

Discretization of Errors

Theorem (Theorem 10.2)

Let $\mathcal{E}$ be a noise process with operation elements $\{E_i\}$, and let $\mathcal{R}$ be a quantum operation that corrects $\mathcal{E}$. If $\mathcal{F}$ is any noise process with operation elements $\{F_j\}$ such that each F_j can be expressed as a linear combination of the E_i:

$$F_j = \sum_i m_{ji} E_i$$

for some complex matrix m_{ji}, then the recovery operation $\mathcal{R}$ also corrects the noise process $\mathcal{F}$.

Implications

- We do not need to correct an infinite number of possible error processes.
  - ◇ So long as a basis satisfies the QEC conditions, we are protected against any arbitrary linear combination of those errors.
- The "analog" problem of continuous quantum noise is now a "digital" problem.
  - ◇ Quantum noise causes continuous rotations on the Bloch sphere by any arbitrary angle, which is a challenge.
  - ◇ Linearity allows to express any arbitrary noise as a linear combination of the Pauli basis $\{I, X, Y, Z\}$.
    - ○ No need to correct the precise angle; we only need to correct the discrete basis errors.
  - ◇ The syndrome measurement forces the continuous noise to collapse into a definite, discrete outcome (e.g., "an X error happened").
  - ◇ Once the error is categorized as a discrete event, the problem becomes finite.
    - ○ We simply apply the corresponding inverse gate to flip the qubit back to its original state.

Proof of Discretization: I. Setup and Assumptions

- $\{E_i\}$ must satisfy the QEC conditions, $PE_i^\dagger E_j P = \alpha_{ij} P$.
  - Without loss of generality, may assume that the basis of errors $\{E_i\}$ satisfies the diagonalized Quantum Error-Correction (QEC) condition:

$$PE_k^\dagger E_i P = \delta_{ki} d_{kk} P$$

- Key properties used in the proof:
  - Polar Decomposition: $E_i P = \sqrt{d_{ii}} U_i P$
  - Error Subspace Projector: $P_k = U_k P U_k^\dagger$
  - Code Space Invariance: Since ρ is in the code space, $P\sqrt{\rho} = \sqrt{\rho}$.

Proof of Discretization: II. Deriving the Recovery Action

- The error-correction operation $\mathcal{R}$ has operation elements $U_k^\dagger P_k$.
- **Key identity:** From $E_k P = \sqrt{d_{kk}} U_k P$, taking the adjoint gives $P E_k^\dagger = \sqrt{d_{kk}} P U_k^\dagger$, so $P U_k^\dagger = \frac{1}{\sqrt{d_{kk}}} P E_k^\dagger$.
- Evaluate the action of the recovery operator $U_k^\dagger P_k$ on an error E_i acting on a state $\sqrt{\rho}$:

$$\begin{aligned} U_k^\dagger P_k E_i \sqrt{\rho} &= U_k^\dagger (U_k P U_k^\dagger) E_i P \sqrt{\rho} = P U_k^\dagger E_i P \sqrt{\rho} \\ &= \frac{1}{\sqrt{d_{kk}}} (P E_k^\dagger E_i P) \sqrt{\rho} \qquad \text{(Key identity above)} \\ &= \delta_{ki} \sqrt{d_{kk}} \sqrt{\rho}. \qquad \text{(Diagonal QEC condition)} \end{aligned}$$

- Interpretation:
  - If $k \neq i$: The recovery channel for k ignores the error i.
  - If $k = i$: The expression is $\sqrt{d_{kk}}\sqrt{\rho}$. The error is perfectly undone, leaving the original state scaled by the error probability.

Proof of Discretization: III. Correcting the New Noise

- Substituting $F_j = \sum_i m_{ji} E_i$ gives:

$$U_k^\dagger P_k F_j \sqrt{\rho} = \sum_i m_{ji} \delta_{ki} \sqrt{d_{kk}} \sqrt{\rho}$$
$$= m_{jk} \sqrt{d_{kk}} \sqrt{\rho}.$$

- Applying the recovery operation $\mathcal{R}$ to the new noise process $\mathcal{F}$:

$$\mathcal{R}(\mathcal{F}(\rho)) = \sum_{k,j} U_k^\dagger P_k F_j \rho F_j^\dagger P_k U_k$$
$$= (\sum_{k,j} |m_{jk}|^2 d_{kk}) \rho \propto \rho.$$

- Because the output is directly proportional to the original state ρ, the error-correction operation $\mathcal{R}$ successfully corrects the arbitrary noise process $\mathcal{F}$, as required.

Limits of QEC: The Quantum Hamming Bound

- The big question: What is the minimum number of physical qubits n needed to protect k logical qubits from t errors?
  - ◇ In classical digital codes, every unique bit-flip error creates a distinct corrupted string.
    - ○ Need enough extra bits to act as unique addresses to identify every possible error.
  - ◇ For n physical qubits, an arbitrary single-qubit error can be an X, Y, or Z fault. Thus, for any specific number of errors j $(0 \leq j \leq t)$, there are $\binom{n}{j}3^j$ possible error combinations.
  - ◇ Total number of distinct error operators the code must handle:

$$\sum_{j=0}^{t} \binom{n}{j} 3^j.$$

Hamming Bound: Non-Degenerate Codes

- A non-degenerate code requires that:
  - ◇ Every distinct error must map the code space to a subspace that is completely orthogonal to the original code space.
  - ◇ Every distinct error must map to a subspace that is completely orthogonal to every other error's subspace.
- A system of k qubits requires a subspace of dimension 2^k.
- Each error acting on the code space copies the entire 2^k dimension into a brand-new, independent pocket of the physical Hilbert space.
  - ◇ The total dimension consumed by the uncorrupted code space and all its error variations is

$$\sum_{j=0}^{t} \binom{n}{j} 3^j 2^k \leq 2^n$$

  - ◇ To protect $k = 1$ qubit from $t = 1$ error, the bound dictates $2(1 + 3n) \leq 2^n$. The smallest integer that satisfies this is $n = 5$.

The Quantum Loophole: Degenerate Codes

- The quantum Hamming bound assumes every error must be uniquely identifiable.
- Quantum mechanics allows a fascinating exception:
  - A code is said to be degenerate if distinct physical errors have the exact same effect on the encoded logical state.
    - The Shor 9-qubit Code: A phase flip (Z) on qubit 1 and a phase flip (Z) on qubit 2 produce identical error syndromes.
  - Because the recovery operation for both errors is identical, the syndrome measurement does not need to distinguish between them. Nature collapses them into the same error class.
- Takeaway for hardware design:
  - Because multiple physical errors land in the same subspace, the number of error subspaces is strictly less than the number of physical errors.
  - Opens up a theoretical pathway to engineer highly efficient, compact quantum codes that protect against a large number of physical errors using far fewer physical qubits (n) than a non-degenerate code would strictly demand.

Classical Linear Codes: Algebraic Structure

Definition (Linear Code)

A binary **linear code** $\mathcal{C}$ of length n and dimension k is a k-dimensional vector subspace of $\mathbb{F}_2^n$.

- **Subspace Property:** Closure under bitwise XOR ($x, y \in \mathcal{C} \implies x \oplus y \in \mathcal{C}$) and contains the all-zero vector $\mathbf{0} \in \mathcal{C}$.
- **Code Parameters $[n, k, d]$:**
  - n: Total block length (physical bits).
  - k: Number of encoded message bits ($|\mathcal{C}| = 2^k$ valid codewords).
  - d: Minimum Hamming distance of the code.
- **Dual Matrix Representations:** (More details later!)
  - **Generator Matrix ($G \in \mathbb{F}_2^{n \times k}$):** Encodes messages via $\mathcal{C} = \{Gx \mid x \in \mathbb{F}_2^k\}$.
  - **Parity-Check Matrix ($H \in \mathbb{F}_2^{(n-k) \times n}$):** Validates codewords via $\mathcal{C} = \{c \in \mathbb{F}_2^n \mid Hc = \mathbf{0}\}$.

Distance Metric & The Linear Code Shortcut

- **Hamming Metrics:**
  - Distance $d(x, y)$: Number of bit flips required to convert codeword x into y.
  - Weight $\text{wt}(x)$: Number of non-zero entries (1s) in a codeword x.
- **The Subspace Equivalence:** For any pair $x, y \in \mathcal{C}$, the difference string $(x \oplus y) = z$ is itself a valid codeword in $\mathcal{C}$.

$$d(x, y) = \text{wt}(x \oplus y) = \text{wt}(z) = d(z, \mathbf{0})$$

- **Minimum Distance:** Instead of checking all $\binom{2^k}{2}$ pairwise distances, d_{min} reduces to the minimum weight among non-zero codewords:

$$d_{\text{min}} = \min_{x \neq y \in \mathcal{C}} d(x, y) = \min_{x \in \mathcal{C} \setminus \{\mathbf{0}\}} \text{wt}(x)$$

- **Error Correction Capability:** To fix t errors without confusion, the 'error balls' of radius t around two codewords must not overlap. Thus, the total distance between codewords must be at least $2t + 1$, giving us $t = \lfloor \frac{d-1}{2} \rfloor$.
  - A code with minimum distance d can correct up to $t = \lfloor \frac{d-1}{2} \rfloor$ bit-flip errors.

Classical Linear Code: Generator and Parity Check

- Key idea: Linearity allows us to use standard matrix operations over $\mathbb{F}_2$ to encode data and detect errors efficiently.
- Encoding: An $n \times k$ generator matrix G maps a message x to a codeword c:

$$c = Gx$$

- Error detection (H): An $(n-k) \times n$ parity check matrix H defines the code as its kernel.
  - A vector c is a valid codeword if and only if:

$$Hc = 0$$

  - If noise adds an error vector e (so we receive $y = c + e$), compute the syndrome: $Hy = H(c + e) = He$.

An Example

- The three-parameter $[n, k, d]$ notation is standard practice in coding.
  - ◇ It costs n physical bits to send the data.
  - ◇ It carries k actual bits of information.
  - ◇ It has enough geometric separation to fix any t bit errors provided $d \geq 2t + 1$.
- Consider the $[3, 1, 3]$ repetition code ($k = 1, n = 3, d = 3$).
  - ◇ Construct the filters:

$$G = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}, \quad H = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix}$$

    - ○ Encoding $x = [1]$ by $c = [1, 1, 1]^T$.
    - ○ Insisting $Hc = 0$. (Check $c_1 = c_2$ and $c_2 = c_3$.)
  - ◇ If the second bit flips, we receive $y = [1, 0, 1]^T$. The syndrome is $Hy = [1, 1]^T \neq 0$, uniquely pointing to an error on bit 2.

Calderbank-Shor-Steane (CSS) Codes: Motivation

- The difficulty in quantum systems:
  - Two independent types of discrete errors occur simultaneously: bit flips (X) and phase flips (Z).
  - Designing a quantum code from scratch to handle both is difficult because of entanglement.
    - Because X and Z anticommute, a naive X-syndrome measurement will disturb and collapse the phase information we are trying to protect, and vice versa!
- The basic idea behind CSS codes:
  - Take two separate, highly optimized classical linear codes; one designed to fix classical bit flips, and another to fix classical phase flips.
  - Glue them together into a single quantum code.

Transition to CSS Codes: Scaling Beyond Shor

- **Beyond Shor's $[9, 1, 3]$ Code:**
  - Shor's 9-qubit code proved QEC is possible by combining two classical repetition codes—one for bit flips ($0 \to 000$) and one for phase flips ($+ \to +++$).
  - **The Limitation:** It is ad-hoc and inefficient—requiring 9 physical qubits to protect a single logical qubit against 1 error.
- **The CSS Framework (Calderbank–Shor–Steane):**
  - Generalizes Shor's intuition into a **systematic recipe**: construct quantum codes directly from two classical linear codes (C_1 for X-errors, C_2 for Z-errors).
- **Why are CSS Codes the gold standard:**
  - **Imports 50+ Years of Classical Theory:** Allows us to build quantum codes directly from classical Hamming, Reed–Muller, and LDPC codes.
  - **Higher Efficiency:** Unlocks codes like the $[7, 1, 3]$ Steane code, achieving Shor's error protection with 2 fewer qubits.
  - **Solves Anti-Commutation:** The simple algebraic condition $C_2^{\perp} \subset C_1$ mathematically guarantees that X and Z checks commute.

Calderbank-Shor-Steane (CSS) Codes: Ingredients

- Let C_1 be an $[n, k_1]$ classical code to handle X errors, and C_2 be an $[n, k_2]$ classical code chosen such that $C_2 \subset C_1$ (nested subcode).
- Partition C_1 into distinct cosets $x + C_2$.
  - ◇ The number of unique cosets is exactly:

$$\frac{|C_1|}{|C_2|} = \frac{2^{k_1}}{2^{k_2}} = 2^{k_1 - k_2}.$$

- Each independent coset corresponds to exactly one logical quantum state.
  - ◇ The resulting quantum code encodes exactly $k_1 - k_2$ logical qubits.

Calderbank-Shor-Steane (CSS) Codes: Construction

- For every valid classical coset representative $x \in C_1/C_2$, define a logical quantum basis state as a uniform superposition:

$$|x + C_2\rangle \equiv \frac{1}{\sqrt{|C_2|}} \sum_{y \in C_2} |x + y\rangle$$

- The role of superposition:
  - ◇ Bit-Flip Protection (X): Handled by the outer code C_1.
  - ◇ Phase-Flip Protection (Z): Handled by the dual code $C_2^{\perp}$.
    - ○ The equal superposition of states within the sum causes phase errors to register as simple bit-shifts in the dual basis, making them classically detectable.
  - ◇ Quantum decoding is split cleanly into two independent classical decoding steps.

CSS Explained: I. The Fundamental Challenge of QEC

- In classical coding, we take a received vector y and compute the check Hy.
  - ◇ If $Hy \neq 0$, the resulting vector tells exactly which bit is flipped, which can be flipped back.
- In quantum mechanics, we cannot use this direct approach:
  - ◇ We are **not allowed** to read the individual physical qubits to see if they hold a 0 or a 1.
  - ◇ Direct observation causes *wavefunction collapse*, instantly destroying the delicate superposition containing our data.
- The strategy is to extract the error identity (the syndrome) as an eigenvalue of a joint operator, leaving the data superposition completely undisturbed.

CSS Explained: II. Relate to Classical Checks

- Consider a repetition code: $h = (1, 1, 0)$.
  - The classical parity checks if the first bit equals the second bit ($c_1 + c_2 = 0$).
- Map this classical check into a quantum operator using the diagonal Pauli Z matrix and the Identity matrix I.
  - Use this quantum check operator (stabilizer) to match the row h:

$$M_1 = Z \otimes Z \otimes I.$$

  - Performs exact the same classification on a quantum state, but yields the error information as an eigenvalue (± 1) instead of a binary bit (0 or 1).

CSS Explained: III. Tracking Bit-Flip (X)

- Suppose the pristine encoded logical state is:

$$|\psi\rangle = \frac{1}{\sqrt{2}}\big(|000\rangle + |111\rangle\big).$$

- Suppose noise causes an X (bit-flip) error on the second qubit. The original state vector is hit to become:

$$|\psi_{\text{broken}}\rangle = \frac{1}{\sqrt{2}}\big(|010\rangle + |101\rangle\big).$$

- How does the quantum check operator $M_1 = Z \otimes Z \otimes I$ respond to this corrupted state?

CSS Explained: III. Tracking Bit-Flip (X): The Math

- Apply M_1 to each term inside the broken state vector:

$$M_1 |010\rangle = (Z|0\rangle) \otimes (Z|1\rangle) \otimes (I|0\rangle) = (+1)(-1)(+1)|010\rangle = -1|010\rangle ,$$
$$M_1 |101\rangle = (Z|1\rangle) \otimes (Z|0\rangle) \otimes (I|1\rangle) = (-1)(+1)(+1)|101\rangle = -1|101\rangle .$$

- Recombining the terms:

$$M_1 |\psi_{\text{broken}}\rangle = -1 \left[\frac{1}{\sqrt{2}} (|010\rangle + |101\rangle) \right] = -1 |\psi_{\text{broken}}\rangle .$$

  - ◇ The state vector itself was not collapsed or altered, but it yielded an eigenvalue of -1, flagging a mismatch between qubits 1 and 2!
  - ◇ **Crucial point:** The measurement did *not* collapse the logical amplitudes α and β. Both $|010\rangle$ and $|101\rangle$ yielded the *same* eigenvalue -1, so the superposition is preserved. The eigenvalue encodes the *error location*, not the data.

CSS Explained: IV. Tracking Phase-Flip (Z)

- A phase-flip error alters the relative sign (phase) inside the superposition:

$$\frac{1}{\sqrt{2}}(|000\rangle + |111\rangle) \longrightarrow \frac{1}{\sqrt{2}}(|000\rangle - |111\rangle)$$

- Construct the X-type check operator $M_2 = X \otimes X \otimes X$ and apply it explicitly:

$$M_2 \cdot \tfrac{1}{\sqrt{2}}(|000\rangle - |111\rangle) = \tfrac{1}{\sqrt{2}}(|111\rangle - |000\rangle) = -\tfrac{1}{\sqrt{2}}(|000\rangle - |111\rangle).$$

  - The phase-flipped state is an eigenstate of M_2 with eigenvalue $-\mathbf{1}$, flagging the error cleanly.
  - The un-flipped state $\frac{1}{\sqrt{2}}(|000\rangle + |111\rangle)$ gives eigenvalue $+1$. The syndrome perfectly distinguishes the two without disturbing the logical amplitudes.

CSS Code: Unification

- A CSS code is a masterfully engineered architecture that unifies these two parallel detection routines:
  - ◇ It builds a set of pure Z-type check operators from a classical code matrix to hunt down physical bit flips.
  - ◇ It builds a separate set of pure X-type check operators from another classical code matrix to hunt down physical phase flips.
- Why subcode nesting ($C_2 \subset C_1$) is mandatory:
  - ◇ Guarantees that every single X-type check operator commutes with every single Z-type check operator ($[M_X, M_Z] = 0$).
  - ◇ Because they commute, we can run both matrix checks simultaneously on the state without them interfering with or scrambling each other's results.
- CSS codes form the mathematical foundation for modern applications such as surface codes and toric codes used in experimental superconducting processors.
- Note: The check operators M_Z and M_X here are the first examples of **stabilizers** — the next section develops this into a full algebraic framework.

The Steane Code: Classical Example

- The Steane code is the canonical example of a CSS code.
  - ◇ Use the classical $[7, 4, 3]$ Hamming code as its primary building block.
  - ◇ The Hamming code C_1 is a 4-dimensional subspace.
  - ◇ The columns of H must consist of every non-zero binary vector of length $m = 3$. Therefore, can write H as

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

    - ○ Crucial algebraic properties of H (mod 2):

$$HH^T = 0 \pmod 2.$$

  - ◇ The code space C_1 is defined geometrically as the null space of H:

$$C_1 = \ker(H) = \{c \in \mathbb{F}_2^7 \mid Hc = 0\}$$

The Steane Code: Meeting the CSS Condition

- A CSS code must satisfy the nested subcode structure: $C_2 \subset C_1$.
- Choose $C_2 = C_1^{\perp}$.
  - The rows of H form the basis vectors for $C_1^{\perp}$.
  - Because $HH^T = \mathbf{0}$, every basis vector of $C_1^{\perp}$ satisfies the parity checks of C_1.
  - This proves that $C_1^{\perp} \subset C_1$, satisfying the nesting constraint seamlessly.

The Steane Code: Qubit & Distance Counting

- Count the parameters of the resulting quantum code:
  - ◇ Physical Qubits: Both codes use 7-bit words $\implies$ $n = 7$.
  - ◇ Dimension of C_1: The code space dimension is $k_1 = 4$.
  - ◇ Dimension of C_2 (k_2): Since $C_2 = C_1^{\perp}$, its dimension is $k_2 = 3$.
  - ◇ Logical Qubits (k): $k = k_1 - k_2 = 4 - 3 = 1$.
- Because both classical components have a minimum distance of 3, the Steane code is a $[7, 1, 3]$ quantum code that can correct any single arbitrary qubit error.

The Steane Code: Explicit Logical States

- The logical basis states are constructed by summing uniformly over the entries of the subcode $C_2 = C_1^{\perp}$.
- The dual Hamming code contains $2^3 = 8$ vectors, consisting of the all-zero vector and seven vectors of weight 4. These are the 8 even-weight codewords:

$$|0_L\rangle = \frac{1}{\sqrt{8}} \sum_{y \in C_1^{\perp}} |y\rangle$$

- To construct $|1_L\rangle$, we shift the entire coset by adding the all-ones codeword $\mathbf{1} = [1,1,1,1,1,1,1]^T \in C_1$. This yields the remaining 8 odd-weight codewords:

$$|1_L\rangle = \frac{1}{\sqrt{8}} \sum_{y \in C_1^{\perp}} |y + \mathbf{1}\rangle$$

Decoding the $[7, 4, 3]$ Label: The Matrix Blueprint

- The parameters $[n, k, d] = [7, 4, 3]$ dictate a linear vector space structure over the finite field $\mathbb{F}_2$:
  - $n = 7$) means that the codewords are vectors of length 7.
  - $k = 4$ means that the code C_1 is a 4-dimensional subspace, thus must contain exactly $2^4 = 16$ distinct codewords.
  - Parity constraints $m = n - k = 3$ means that there are 3 independent linear equations, forming a 3×7 parity-check matrix H.
- The parity-check means that the code space C_1 is defined geometrically as the kernel (null space) of H.

Finding the Subspace C_1: The Generator Matrix

- Standard linear algebra technique.
- Solving the homogeneous system $Hc = 0$ yields 4 free variables.
  - ◇ Setting each free variable to 1 and the rest to 0 defines the 4×7 Generator Matrix G, whose rows form the basis $\{\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3, \mathbf{v}_4\}$:

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

  - ◇ Every valid classical codeword is generated via linear combinations modulo 2:

$$\mathbf{c} = \alpha_1 \mathbf{v}_1 + \alpha_2 \mathbf{v}_2 + \alpha_3 \mathbf{v}_3 + \alpha_4 \mathbf{v}_4, \quad \alpha_i \in \{0, 1\}$$

- Evaluating all $\binom{4}{0} + \binom{4}{1} + \binom{4}{2} + \binom{4}{3} + \binom{4}{4}$ row combinations systematically rolls out the entire roster of 16 codewords.
- The total span splits evenly into 8 even-weight vectors (C_2) and 8 odd-weight vectors (the coset $\mathbf{1} + C_2$).

The Pauli Group

- The Pauli group G_n on n qubits consists of the set of all n-qubit Pauli strings under the operation of standard matrix multiplication.
  - ◇ The Pauli matrices satisfy the relationships

$$XX = YY = ZZ = -\imath XYZ = I. \tag{1}$$

    - ○ Multiplying any two Pauli strings yields another Pauli string multiplied by a phase factor $\in \{\pm 1, \pm i\}$.
    - ○ For any element $g = \omega \cdot \left(\bigotimes P_i\right)$, its matrix inverse is simply $g^{-1} = \omega^{-1} \cdot \left(\bigotimes P_i\right)$.
    - ○ For any two operators $P, Q \in \mathcal{G}_n$, they satisfy:

$$PQ = (-1)^s QP \quad \text{where } s \in \{0, 1\}. \tag{2}$$

- The single-qubit Pauli group has 16 elements:

$$G_1 = \left\{ \pm I, \pm iI, \pm X, \pm iX, \pm Y, \pm iY, \pm Z, \pm iZ \right\}.$$

The Stabilizer Formalism: A Precise Group Action

- Let the Pauli group G_n act on the n-qubit Hilbert space $\mathcal{H} = (\mathbb{C}^2)^{\otimes n}$ via standard matrix-vector multiplication: $g \cdot |\psi\rangle = g\,|\psi\rangle$.
- A **stabilizer** S is an abelian subgroup of G_n that isolates a invariant subspace $V_S \subseteq \mathcal{H}$ defined by:

$$V_S = \{|\psi\rangle \in \mathcal{H} \mid g\,|\psi\rangle = +1\,|\psi\rangle\,, \quad \forall g \in S\}.$$

  - V_S is the simultaneous $+1$ eigenspace of all operators in S.
  - $\dim(V_S) = 2^{n-\dim(S)}$.
- Necessary conditions for a non-trivial code space ($V_S \neq \{0\}$):
  - Elements of S must commute ($[g_i, g_j] = 0$).
    - Non-commuting operators cannot share a simultaneous eigenspace.
  - $-I \notin S$.
    - If $-I \in S$, then $-I\,|\psi\rangle = |\psi\rangle \implies |\psi\rangle = \mathbf{0}$, collapsing the subspace.

The Binary Symplectic Representation

- To analyze stabilizer groups using linear algebra, we map n-qubit Pauli strings to binary vectors of length $2n$ by stripping away global phases.
- For single-qubit mapping: Every Pauli matrix can be written as $X^x Z^z$ for $x, z \in \{0, 1\}$. We map this to a coordinate pair $(x \mid z) \in \mathbb{F}_2^2$:

$$I \to (0 \mid 0), \quad X \to (1 \mid 0), \quad Z \to (0 \mid 1), \quad Y \to (1 \mid 1)$$

- For n-qubit extension: For an n-qubit string, we concatenate the X-components and Z-components into a single row vector $\mathbf{r} = (\mathbf{x} \mid \mathbf{z}) \in \mathbb{F}_2^{2n}$:

$$g = P_1 \otimes P_2 \otimes \cdots \otimes P_n \longrightarrow \mathbf{r} = (x_1, \ldots, x_n \mid z_1, \ldots, z_n)$$

  - ◇ An example: For $g = X \otimes Y \otimes Z$ on 3 qubits:

$$\mathbf{x} = (1, 1, 0), \quad \mathbf{z} = (0, 1, 1) \implies \mathbf{r} = (1, 1, 0 \mid 0, 1, 1)$$

Constructing the Stabilizer Check Matrix

- If a stabilizer subgroup S has ℓ independent generators $\{g_1, \ldots, g_\ell\}$, stacking their binary row vectors yields an $\ell \times 2n$ Check Matrix $H =:$

$$H = (X \mid Z) := \left(\begin{array}{ccc|ccc} x_{1,1} & \cdots & x_{1,n} & z_{1,1} & \cdots & z_{1,n} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ x_{l,1} & \cdots & x_{l,n} & z_{l,1} & \cdots & z_{l,n} \end{array}\right)$$

- The algebraic payoff:
  - ◇ Operator multiplication inside $\mathcal{G}_n$ becomes vector addition in $\mathbb{F}_2^{2n}$.
  - ◇ **Commutativity $\leftrightarrow$ symplectic orthogonality:** For $P = X^{x_1} Z^{z_1}$, $Q = X^{x_2} Z^{z_2}$, using $XZ = -ZX$:

$$PQ = (-1)^{z_1 \cdot x_2} X^{x_1 + x_2} Z^{z_1 + z_2}, \quad QP = (-1)^{x_1 \cdot z_2} X^{x_1 + x_2} Z^{z_1 + z_2}.$$

    So P, Q commute $\iff$ $\mathbf{x}_1 \cdot \mathbf{z}_2 + \mathbf{z}_1 \cdot \mathbf{x}_2 \equiv 0 \pmod 2$.
    - ◦ The required group property that all generators must commute reduces to the matrix constraint:

$$H \Omega H^T = \mathbf{0} \pmod 2, \quad \text{where } \Omega = \begin{pmatrix} \mathbf{0} & I_n \\ I_n & \mathbf{0} \end{pmatrix}.$$

Generators and Check Matrices

- **Proposition 10.5:** If S is generated by ℓ independent and commuting elements and $-I \notin S$, then the stabilized code space V_S is a 2^k-dimensional vector space, where $k = n - \ell$.
- **Proof sketch:**
  - ◇ Each independent stabilizer generator g_i is a Hermitian operator with eigenvalues ± 1, so it splits the ambient space into two equally-sized eigenspaces.
  - ◇ Restricting to the $+1$ eigenspace halves the dimension: $2^n \to 2^{n-1}$.
  - ◇ Since the ℓ generators are independent and mutually commuting, each successive restriction halves the dimension independently:

$$2^n \xrightarrow{g_1} 2^{n-1} \xrightarrow{g_2} 2^{n-2} \xrightarrow{\cdots} 2^{n-\ell} = 2^k.$$

  - ◇ We start with n physical qubits (2^n dimensions) and use ℓ stabilizer checks to protect $k = n - \ell$ logical qubits (2^k dimensions).

Why the Pauli Group? The Measurement Paradox

- The fundamental paradox of QEC:
  - Detect and correct errors within the logical state $|\psi\rangle = \alpha|0_L\rangle + \beta|1_L\rangle$.
  - Cannot look at the state as it will collapse the superposition.
- The stabilizer formalism solves this by exploiting the discrete algebraic structure of the Pauli group.
  - Using stabilizer subgroups from the Pauli group bypasses this entirely through two elegant algebraic mechanisms: non-destructive syndrome extraction and error discretization.

Why the Pauli Group? Syndrome Extraction

- Let $|\psi\rangle \in V_S$ be the pristine code state.
  - ◇ By definition, $g_i |\psi\rangle = +1 |\psi\rangle$ for all $g_i \in S$.
- Suppose a physical Pauli error $E \in G_n$ occurs.
  - ◇ The state is corrupted to $E |\psi\rangle$.
- Because $g_i, E \in G_n$, they must either commute ($[g_i, E] = 0$) or anti-commute ($\{g_i, E\} = 0$).
  - ◇ Upon measuring the stabilizer observable g_i on the corrupted state :

$$g_i(E |\psi\rangle) = \begin{cases} +1(E |\psi\rangle) & \text{if } [g_i, E] = 0 \\ -1(E |\psi\rangle) & \text{if } \{g_i, E\} = 0 \end{cases}$$

- The binary eigenvalue (± 1) reveals information about the error relationship without revealing any information about the coefficients α and β.

Why the Pauli Group? Error Discretization (Part 1)

- Physical environmental noise is continuous and analog, not discrete.
  - ◇ Suppose a continuous rotation error occurs as

$$|\Phi\rangle = \cos(\theta)\,|\psi\rangle + i\sin(\theta)X\,|\psi\rangle\,.$$

- Measure the stabilizer observable g_i that anti-commutes with X:
  - ◇ The spectral decomposition of g_i is

$$g_i = (+1)P_+ + (-1)P_- \implies g_i = P_+ - P_-$$

    with the only two eigenvalues $+1$ and -1.
  - ◇ Can write the orthogonal projection operators as

$$P_+ = \frac{I + g_i}{2} \quad \text{and} \quad P_- = \frac{I - g_i}{2}.$$

  - ◇ Note the action is to take "measurement," which gives rise to only two outcomes.

Why the Pauli Group? Error Discretization (Part 2)

- Case 1: Projecting onto the $+1$ Eigenspace:
  - Apply P_+ to the corrupted state $|\Phi\rangle$:

$$\begin{aligned} P_+ |\Phi\rangle &= \frac{I + g_i}{2}\Big(\cos(\theta)|\psi\rangle + i\sin(\theta)X|\psi\rangle\Big) \\ &= \frac{1}{2}\Big[\cos(\theta)|\psi\rangle + i\sin(\theta)X|\psi\rangle + \cos(\theta)\mathbf{g_i}|\psi\rangle + i\sin(\theta)\mathbf{g_i}\mathbf{X}|\psi\rangle\Big] \\ &= \frac{1}{2}\Big[\cos(\theta)|\psi\rangle + i\sin(\theta)X|\psi\rangle + \cos(\theta)|\psi\rangle - i\sin(\theta)X|\psi\rangle\Big] \\ &= \cos(\theta)|\psi\rangle \end{aligned}$$

- The result;
  - Probability of $+1$ outcome is $\|P_+|\Phi\rangle\|^2 = \cos^2(\theta)$.
  - Post-measurement normalized state becomes $\frac{\cos(\theta)|\psi\rangle}{\cos(\theta)} = |\psi\rangle$.
  - Final verdict: The continuous noise is completely erased.

Why the Pauli Group? Error Discretization (Part 3)

- Case 2: Projecting onto the -1 Eigenspace:
  - Apply P_- to the corrupted state $|\Phi\rangle$:

$$\begin{aligned} P_- |\Phi\rangle &= \frac{I - g_i}{2}\Big(\cos(\theta)|\psi\rangle + i\sin(\theta)X|\psi\rangle\Big) \\ &= \frac{1}{2}\Big[\cos(\theta)|\psi\rangle + i\sin(\theta)X|\psi\rangle - \cos(\theta)|\psi\rangle + i\sin(\theta)X|\psi\rangle\Big] \\ &= i\sin(\theta)X|\psi\rangle\,. \end{aligned}$$

- The result:
  - Probability of -1 outcome is $\|P_-|\Phi\rangle\|^2 = \sin^2(\theta)$.
  - Post-measurement normalized state becomes $\frac{i\sin(\theta)X|\psi\rangle}{\sin(\theta)} \sim X|\psi\rangle$.
  - Final verdict:
    - The undamaged $\cos(\theta)|\psi\rangle$ cancel out completely
    - The continuous noise is forced into a discrete, clean Pauli X error.

What if the Stabilizer Commutes with the Error?

- Suppose the stabilizer generator g_i **commutes** with the noise operator X
- It can be checked that applying $P_+ = \frac{I+g_i}{2}$ and exploiting the commuting property leads to

$$P_+ |\Phi\rangle = |\Phi\rangle.$$

- The result is that g_i is entirely blind to this noise channel.
  - Probability of $+1$ outcome is $\| P_+ |\Phi\rangle \|^2 = 1$.
  - Probability of -1 outcome is $\| P_- |\Phi\rangle \|^2 = 0$.
  - The continuous analog error passes through completely undetected and un-digitized.

What If the Noise Is in Other Directions (X, Y, or Z)

- Environmental noise is rarely just a bit-flip (X).
  - ◇ Model the continuous noise using a Pauli string $E \in \{X, Y, Z\}^{\otimes n}$:

$$E_{\text{continuous}} = \cos(\theta) I + i \sin(\theta) E$$

- Because of the rigid binary structure of the Pauli group, the stabilizer g_i treats any Pauli noise string E via a strict geometric dichotomy:

Case A: $[g_i, E] = 0$

Commuting Noise

- ◇ Measurement outcome is $+1$ with 100% certainty.
- ◇ The continuous noise passes through completely unchanged.

Case B: $\{g_i, E\} = 0$

Anti-commuting Noise

- ◇ Continuous noise is collapsed.
- ◇ Prob $\cos^2\theta$: Erased ($|\psi\rangle$).
- ◇ Prob $\sin^2\theta$: Digitized into a clean discrete Pauli error ($E|\psi\rangle$).

- The discretization mechanism does not care about the labels X, Y, or Z. It cares exclusively about commutation.

The Big Picture: Teamwork

- The three roles of a single stabilizer combine synergistically across a full generator set.
- The setup:
  - ◇ A stabilizer $S = \langle g_1, g_2, \ldots, g_\ell \rangle$ with generators engineered so that every dangerous Pauli error E anti-commutes with a **unique combination** of generators.
- The work:
  - ◇ Measuring all ℓ generators yields an ℓ-bit syndrome $(\pm 1, \ldots, \pm 1)$.
  - ◇ An error commuting with g_1 but anti-commuting with g_2: g_1 is blind, g_2 digitises and flags it.
- Together, the full team maps out the exact locations of physical defects across the Hilbert space.

Synthesis: The Three Roles of Stabilizers, I

- Stabilizers $g \in S$ do not act as shields that repel or absorb noise. Instead, they serve three geometric and information-theoretic roles.
  - ◇ Defining subspaces via Invariants (structural function):
    - ○ The 2^k-dimensional codespace V_S is not tracked explicitly by its massive basis vectors.
    - ○ Rather, it is defined implicitly by its invariants:

$$V_S = \{\, |\psi\rangle \in \mathcal{H}^{\otimes n} : g_i\,|\psi\rangle = +1\,|\psi\rangle\,,\ \forall g_i \in S\}.$$

    - ○ Each independent generator cuts the ambient Hilbert space exactly in half.
  - ◇ Non-destructive error alarms (operational function):
    - ○ Pauli noise E either commutes or anti-commutes with a stabilizer g.
    - ○ Evaluating the corrupted state yields a clean binary diagnosis:

$$g\big(E\,|\psi\rangle\,\big) = \begin{cases} +Eg\,|\psi\rangle = +1\big(E\,|\psi\rangle\,\big) & \text{if } [g,E]=0 \\ -Eg\,|\psi\rangle = -1\big(E\,|\psi\rangle\,\big) & \text{if } \{g,E\}=0 \end{cases}$$

    - ○ Measuring the observable g extracts an error syndrome (± 1) without collapsing or leaking the encoded logical data superposition.

Synthesis: The Three Roles of Stabilizers, II

- Discretization of continuous drifts (algebraic function)
  - ◇ Real-world environmental noise is typically analog and continuous. A stray magnetic field causing a small phase rotation θ:

$$U_{\text{noise}} = e^{i\theta Z} = \cos(\theta) I + i \sin(\theta) Z,$$

    which creates a coherent superposition: $\cos(\theta) |\psi\rangle + i \sin(\theta) Z |\psi\rangle$.
  - ◇ Measuring a stabilizer g that anti-commutes with Z forces this continuous state to project into discrete alternatives:
    - ○ With probability $\cos^2(\theta)$: Outcome $+1 \implies$ State collapses back to $|\psi\rangle$. The error is actively projected out of existence.
    - ○ With probability $\sin^2(\theta)$: Outcome $-1 \implies$ State collapses to $Z |\psi\rangle$. The error is digitized into a clean, trackable Pauli operator.
  - ◇ Stabilizer measurements act as an information-theoretic prism: they force vague, analog environmental noise to snap into discrete, digital Pauli errors that we can easily fix.

Unitary Evolution of Stabilizer Spaces

Theorem: Subspace Isomorphism

Let g be a stabilizer generator and $g' = UgU^\dagger$ be its evolution under a unitary U. The transformation U maps the $+1$ eigenspace of g bijectively to the $+1$ eigenspace of g':

$$g\left|\psi\right\rangle = \left|\psi\right\rangle \quad \Longleftrightarrow \quad g'\left|\phi\right\rangle = \left|\phi\right\rangle \quad \text{where } \left|\phi\right\rangle = U\left|\psi\right\rangle$$

- Sufficiency: Assume $g\left|\psi\right\rangle = \left|\psi\right\rangle$. Then

$$g'\left|\phi\right\rangle = \left(UgU^\dagger\right)\left(U\left|\psi\right\rangle\right) = Ug\left|\psi\right\rangle = U\left|\psi\right\rangle = \left|\phi\right\rangle.$$

- Necessity: Let $\left|\phi\right\rangle$ be any arbitrary state such that $g'\left|\phi\right\rangle = \left|\phi\right\rangle$. Construct $\left|\psi\right\rangle := U^\dagger\left|\phi\right\rangle$. Then

$$g\left|\psi\right\rangle = g\left(U^\dagger\left|\phi\right\rangle\right) = \left(U^\dagger U\right)gU^\dagger\left|\phi\right\rangle = U^\dagger g'\left|\phi\right\rangle = U^\dagger\left|\phi\right\rangle = \left|\psi\right\rangle.$$

The Clifford Group (The Normalizer)

- To ensure that the transformed stabilizer $UgU^\dagger$ stays in the n-qubit Pauli Group G_n, the unitary U must belong to a very special algebraic structure.
  - ◇ The Clifford Group C_n is the normalizer of the Pauli Group G_n within the group of all unitary matrices $U(2^n)$:

$$C_n = N(G_n) = \{U \in U(2^n) \mid UgU^\dagger \in \mathcal{G}_n,\ \forall g \in \mathcal{G}_n\}$$

- The generators: The Clifford Group is generated entirely by three basic quantum gates: the Hadamard (H), the Phase gate (S), and the Controlled-NOT (CNOT).
- Tracking quantum states under Clifford evolution bypasses exponential matrix-vector multiplication.
  - ◇ Only need to compute how the ℓ generators of S transform under conjugation.

The Operational View: Defining Clifford Gates

- There is a deep algebraic argument behind Clifford generation that transforms the physics of quantum gates into elegant, structured discrete geometry. (Beyond the scope!)
  - ◇ Instead, we define the Clifford group directly by how its gates transform the generators of the Pauli Group via conjugation: $P \mapsto UPU^\dagger$.
- Every Clifford circuit can be compiled from three fundamental gates. Their exact structural actions on the core Pauli operators are:

Clifford Gate (U)	**Action on X ($UXU^\dagger$)**	**Action on Z ($UZU^\dagger$)**
Hadamard (H)	Z	X
Phase (S)	$Y \equiv iXZ$	Z
CNOT$_{1\to2}$ *(1 = Ctrl, 2 = Targ)*	$X \otimes I \to X \otimes X$ $I \otimes X \longrightarrow I \otimes X$	$I \otimes Z$ $Z \otimes Z$

  - ◇ Notice that CNOT propagates X *forward* (control to target) but propagates Z *backward* (target to control).

An Example: Tracking Stabilizer Evolution

- Suppose a quantum state $|\psi\rangle$ is stabilized by $g \in S$. If we apply a Clifford gate U, the new state $U|\psi\rangle$ is stabilized by $g' = UgU^\dagger$.
- Consider the case when an initial 2-qubit state be stabilized by $g = X \otimes I$ and a $\text{CNOT}_{1\to 2}$ gate is applied. What operator stabilizes the output state?
  - Evaluate the conjugation directly by linear transformation of the tensor components:
    $$g' = \text{CNOT}(X \otimes I)\text{CNOT}^\dagger$$
  - Use the table,
    $$g' = X \otimes X.$$
  - Bypass tracking the complex state vector completely.
    - Evolving a stabilizer is reduced to a quick semantic substitution on the string.

The Operational View: The Boundaries of Clifford Circuits

- This operational shortcut yields profound consequences for computational complexity and quantum circuit architecture.
  - ◇ Tracking stabilizer evolution requires only updating a lookup table of n-length Pauli strings.
  - ◇ An n-qubit Clifford circuit containing m gates can be simulated classically in $\mathcal{O}(m \cdot n)$ steps.
- Clifford operations can create massive, multi-qubit entanglement, yet they fail to offer an exponential quantum speedup on their own.
  - ◇ There is a need for non-Clifford resources.
  - ◇ To achieve universal, non-trivial quantum computation, we must introduce at least one non-Clifford element (such as the T-gate, $\theta = \pi/8$ rotation), which breaks this operator-tracking shortcut.

The Gottesman-Knill Theorem

Gottesman-Knill Theorem (1998)

Any quantum circuit operating exclusively with:

1. State preparation in the computational basis ($|0\rangle$)
2. Clifford Group gates (H, S, CNOT) and Pauli gates (X, Y, Z)
3. Measurements of observables in the Pauli Group G_n

can be efficiently simulated on a classical computer in polynomial time $\mathcal{O}(n^2)$.

- Clifford circuits can easily generate highly entangled states (e.g., GHZ states, cluster states).
- Quantum entanglement is necessary, but not sufficient, to achieve exponential quantum speedup.
- To break classical simulability, a quantum computer must utilize non-Clifford elements (such as the T-gate, $\frac{\pi}{8}$-rotation).

The Knill-Laflamme Condition for Stabilizers

Stabilizer Error-Correction Criteria

Let $C(S)$ be a stabilizer code, and let $\mathcal{E} = \{E_j\}$ be a set of physical Pauli errors. The code can protect against all errors in $\mathcal{E}$ if and only if for every pair $E_j, E_k \in \mathcal{E}$:

$$E_j^\dagger E_k \notin N(S) - S$$

- If $E_j^\dagger E_k \in S$, then $E_j^\dagger E_k |\psi\rangle = |\psi\rangle \implies E_k |\psi\rangle = E_j |\psi\rangle$.
  - ◇ The combined operator acts as an identity operator on the code space.
  - ◇ The errors do not need to be distinguished.
- If $E_j^\dagger E_k \notin N(S)$, then it anti-commutes with at least one stabilizer generator g_i.
  - ◇ Guarantees that E_j and E_k will produce distinct binary syndromes.
  - ◇ Allows classical sorting algorithms to distinguish and correct them.

The Failure Mode & Code Distance

- What if $E_j^\dagger E_k \in N(S) - S$?
  - ◇ Such an error operator commutes with every single stabilizer generator.
  - ◇ Since $[E_j^\dagger E_k, g_i] = 0 \ \forall g_i$, the error syndrome yields a perfect $+1$ reading.
    - ○ The syndrome code remains completely blind.
  - ◇ Yet, the operator rotates one valid logical codeword into an entirely different valid logical codeword.
    - ○ The data is corrupted, but undetected.

Code Distance d

- The **weight** of a Pauli operator, $\mathrm{wt}(A)$, is the number of non-identity tensor factors.
- The **distance** d of a stabilizer code is the minimum weight of an undetectable logical error:

$$d = \min_{A \in N(S) - S} \mathrm{wt}(A).$$

- An $[[n, k, d]]$ code detects errors up to weight $d - 1$ and corrects up to weight $t = \lfloor \frac{d-1}{2} \rfloor$.
- **Concrete example — the $[[5, 1, 3]]$ code:**
  - The logical operators $X_L = X^{\otimes 5}$ and $Z_L = Z^{\otimes 5}$ both have weight 5, but *are* in $N(S) - S$.
  - However, one can verify that no weight-1 or weight-2 Pauli string lies in $N(S) - S$.
  - The minimum-weight undetectable logical errors have weight 3, so $d = 3$ and $t = 1$.
  - This confirms the $[[5, 1, 3]]$ code corrects any single-qubit error — matching the Hamming bound.

The Five-Qubit Code: $[5, 1, 3]$

- To protect $k = 1$ logical qubit from an arbitrary single-qubit error ($t = 1$),
  - ◇ Require a minimum code distance of $d = 2(1) + 1 = 3$.
  - ◇ Require $\ell = n - k = 5 - 1 = 4$ independent stabilizer generators.
- Can construct generators via cyclic permutation:
  - ◇ One way to construct the generators for $S = \langle g_1, g_2, g_3, g_4 \rangle$ is by cycling the core string $\mathbf{X} \otimes \mathbf{Z} \otimes \mathbf{Z} \otimes \mathbf{X} \otimes \mathbf{I}$:

$$
\begin{aligned}
g_1 &= X \otimes Z \otimes Z \otimes X \otimes I \\
g_2 &= I \otimes X \otimes Z \otimes Z \otimes X \\
g_3 &= X \otimes I \otimes X \otimes Z \otimes Z \\
g_4 &= Z \otimes X \otimes I \otimes X \otimes Z
\end{aligned}
$$

- (Verify that these generators commute.)

Saturating the Quantum Hamming Bound

- The 5-qubit code is considered a "perfect" code.
  - ◇ It perfectly saturates the spatial limits of quantum mechanics.
  - ◇ Consider the quantum Hamming bound for $t = 1$:
    - ○ For an n-qubit physical space to store k logical qubits while correcting any single-qubit error,

$$\left[\binom{n}{0} + 3\binom{n}{1}\right] 2^k \leq 2^n \implies (1+3n)2^k \leq 2^n$$

    - ○ With the physical parameters $n = 5, k = 1$, the inequality holds as an exact equality:

$$(1 + 3 \times 5) \times 2^1 = 16 \times 2 = 32 = 2^5.$$

  - ◇ Every single coordinate in the 32-dimensional physical Hilbert space is explicitly utilized as either a valid data slot or an error slot.
    - ○ No space is wasted.

Logical Operators for the $[5, 1, 3]$ Code

- Since $\dim(V_S) = 2$, it possesses a logical ground state $|0_L\rangle$ and a logical excited state $|1_L\rangle$.
  - ◇ To manipulate the protected logical information inside S without breaking the code, we must construct logical operations X_L and Z_L.
  - ◇ Properties of logical operators:
    - ○ Must commute with all stabilizer generators ($X_L, Z_L \in N(S)$).
    - ○ Cannot be part of the stabilizer itself ($X_L, Z_L \notin S$).
    - ○ Must satisfy the core Pauli relationship: $\{X_L, Z_L\} = 0$.
- For the 5-qubit code, take symmetric transversals:

$$X_L = X \otimes X \otimes X \otimes X \otimes X$$
$$Z_L = Z \otimes Z \otimes Z \otimes Z \otimes Z$$

  - ◇ X_L anti-commutes with Z_L because it shares 5 slots of anti-commuting XZ operations, and $(-1)^5 = -1$.
  - ◇ Both operators have wt $= 5$, confirming that the logical operations are highly non-local, protecting them from localized physical faults.

Modern Hardware Applications

- The stabilizer formalism is the foundation of modern scalable quantum computing architectures:
  - ◇ The Surface Code: Currently favoured by industry leaders (Google, IBM, Quantinuum). Qubits are arranged on a 2D square lattice with highly localised 4-qubit stabilizers (XXXX on vertices, $ZZZZ$ on faces), enabling rapid, continuous measurement cycles.
  - ◇ Active Syndrome Decoding: The ℓ-bit syndrome vector is fed in real-time to classical graph-theoretic algorithms (Minimum Weight Perfect Matching, Union-Find) to locate and track physical defects.
  - ◇ Transversal Gate Implementations: Stabilizer codes align with the Clifford group, enabling fault-tolerant logical operations (H, S, CNOT) executed component-wise across physical layers.

Fault-Tolerant Quantum Computation

- At its heart, fault-tolerance is not a deep abstraction — it is, at its core, strict quarantine management.
  - The goal is to compute directly on encoded quantum states so that decoding is never required, preventing error accumulation.
- Fault-tolerance means that
  - If only one component fails, the failure causes at most one error in each encoded block of qubits.
  - Each individual block can successfully fix its data if it contains at most one flipped qubit.
  - If a single block ever develops two flipped qubits, it fails catastrophically.
  - This prevents a single error from propagating to multiple qubits within a code block (e.g., via a CNOT gate).
- Encoded gates are automatically fault-tolerant if implemented bitwise across the block.
  - Known as transversality.

Demonstration: 1. The Intra-Block Catastrophe

- Setting:
  - A single $[5, 1, 3]$ code block that can correct at most 1 error.
  - An internal $\text{CNOT}_{1\to 2}$ is executed between two physical qubits inside the same block.
- Consider the state evolution step by step:
  - (Step 1. Ideal state:) The block is completely uncorrupted.

$$|\psi_{\text{start}}\rangle = |00000\rangle$$

  - (Step 2. Component fault:) An X error strikes Qubit 1.

$$|00000\rangle \xrightarrow{X_1 \otimes I^{\otimes 4}} |10000\rangle \implies \text{1 error in block} \quad \text{(Still correctable!!)}$$

  - (Step 3. Gate execution:) The internal $\text{CNOT}_{1\to 2}$ fires. Because the corrupted control (Qubit 1) is $|1\rangle$, it actively flips the target (Qubit 2).

$$|10000\rangle \xrightarrow{\text{CNOT}_{1\to 2}} |11000\rangle \implies \text{2 errors in block} \quad \text{(Catastrophic!!!)}$$

- The single fault propagated inside the same space, breaking the quarantine and overwhelming the code's error-correcting capacity.

Demonstration: 2. Transversal Success (The Quarantine)

- Setting:
  - The system is isolated into two independent 5-qubit blocks (A and B).
  - A transversal (vertical) $\text{CNOT}_{A_1 \to B_1}$ is applied across the blocks.
- Consider the parallel 10-qubit state evolution step by step:
  - (Step 1. Ideal State:) Both 5-qubit blocks are perfectly healthy.

$$|\psi_{\text{start}}\rangle = |00000\rangle_A \otimes |00000\rangle_B$$

  - (Step 2. Component Fault:) An X error strikes Qubit 1 of Block A.

$$|10000\rangle_A \otimes |00000\rangle_B \quad \Longrightarrow \quad \begin{cases} \text{Block A: 1 error} \\ \text{Block B: 0 errors} \end{cases}$$

  - (Step 3. Gate execution:) The transversal $\text{CNOT}_{A_1 \to B_1}$ fires. Qubit A_1 is $|1\rangle$, so it flips Qubit B_1.

$$|10000\rangle_A \otimes |10000\rangle_B \quad \Longrightarrow \quad \begin{cases} \text{Block A: 1 error} & \text{(Correctable!!)} \\ \text{Block B: 1 error} & \text{(Correctable!!)} \end{cases}$$

- The single fault still doubles globally, but is not allowed to multiply locally within a single block. Each block remains at 1 error and can be perfectly repaired.

The No-Go Theorem of Fault-Tolerance

- Question: Cannot we simply implement all gates transversally?

The Eastin-Knill Theorem (2009)

No quantum error-correcting code can implement a universal set of logical gates using exclusively transversal operations.

- The core problem:
  - ◇ The Clifford limitation: Transversal gates on stabilizer codes are strictly confined to the discrete Clifford group ($\langle \text{CNOT}, H, S \rangle$).
  - ◇ The missing piece: To achieve universal quantum computing, we must have a non-Clifford gate, typically the T gate ($\pi/8$ rotation):

$$T = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix}$$

    - ○ $TXT^\dagger = \frac{1}{\sqrt{2}}(X + Y)$, it maps Pauli operators *out* of the Pauli group, destroying the stabilizer structure if applied transversally.

Circumventing Eastin-Knill: Magic State Injection

- **The key insight:** We cannot apply T transversally, but we *can* prepare a special ancilla qubit called a **magic state** and use gate teleportation to transfer its non-Clifford rotation into the data using only Clifford operations.
- The **magic state** is $|T\rangle = \frac{1}{\sqrt{2}}(|0\rangle + e^{i\pi/4}|1\rangle)$.
  - ◇ It is called "magic" because it lies outside the stabilizer state manifold and encodes the non-Clifford rotation $e^{i\pi/4}$ that the T gate applies.
  - ◇ This state is prepared *offline* (using costly but highly reliable distillation protocols), then consumed once to implement one logical T gate.
  - ◇ **Magic state distillation:** Many noisy copies of $|T\rangle$ are purified into fewer, near-perfect copies using Clifford circuits; detailed protocols are discussed in the fault-tolerance literature.

The Injection Procedure (Gate Teleportation)

- Initialization: Combine data $|\psi\rangle_D = \alpha|0\rangle + \beta|1\rangle$ and ancilla $|T\rangle_A$:

$$|\Psi_0\rangle = (\alpha|0\rangle + \beta|1\rangle) \otimes \tfrac{1}{\sqrt{2}}(|0\rangle + e^{i\pi/4}|1\rangle).$$

- Entanglement: Apply CNOT (data=control, ancilla=target). Expanding term by term: CNOT flips the ancilla when the control is $|1\rangle$:

$$\begin{aligned}|\Psi_1\rangle &= \tfrac{\alpha}{\sqrt{2}}|0\rangle_D\Big(|0\rangle + e^{i\pi/4}|1\rangle\Big)_A + \tfrac{\beta}{\sqrt{2}}|1\rangle_D\Big(|1\rangle + e^{i\pi/4}|0\rangle\Big)_A \\ &= \tfrac{1}{\sqrt{2}}\Big[\Big(\alpha|0\rangle_D + \beta e^{i\pi/4}|1\rangle_D\Big)|0\rangle_A + \Big(\alpha e^{i\pi/4}|0\rangle_D + \beta|1\rangle_D\Big)|1\rangle_A\Big].\end{aligned}$$

- Measurement & correction: Measure the ancilla in the Z-basis:
  - Outcome 0: $|\psi_{\text{final}}\rangle = T|\psi\rangle$ (Perfect Success)
  - Outcome 1: $|\psi_{\text{final}}\rangle = e^{i\pi/4}T^\dagger|\psi\rangle \xrightarrow{\text{Apply } S} T|\psi\rangle$ (Clifford Corrected)
  - The S correction ($S = T^2$) is Clifford — fully fault-tolerant.
  - A logical T gate is achieved without ever running a physical non-Clifford gate on the data.

Switching Gears: The Maintenance Problem

- Now that we have a universal logical gate set (Clifford + T), we must address a critical hardware assumption.
  - ◇ How do we actually measure stabilizer syndromes without introducing more errors than the code can handle?
- Why is this a concern:
  - ◇ To measure a multi-qubit stabilizer generator, our ancilla qubits must interact with multiple data qubits. This interaction creates a dangerous pathway for error propagation.

The Naive Approach: The Single-Ancilla Hazard

- Consider measuring the 5-qubit stabilizer $g = X \otimes Z \otimes Z \otimes X \otimes I$.
  - A naive circuit interacts a single ancilla qubit across the data block:

$$\mathrm{CX}_{A\to 1} \longrightarrow \mathrm{CZ}_{A\to 2} \longrightarrow \mathrm{CZ}_{A\to 3} \longrightarrow \mathrm{CX}_{A\to 4}.$$

    - $\mathrm{CX}_{A\to 1}$ means the Ancilla (A) acts as the control, and Data qubit 1 acts as the target for the Pauli-X flip.
- Suppose a single hardware fault (X) occurs on the lone ancilla qubit immediately after the first gate.
  - The X fault sits on the ancilla during $\mathrm{CZ}_{A\to 2}$.
    - $\mathrm{CZ}\,|\Psi\rangle$ is the pristine, ideal state wanted to prepare.
    - The state right before the gate is actually: $(X \otimes I)\,|\Psi\rangle$.
    - Due to the fault,

$$|\Psi_{\text{actual}}\rangle = \mathrm{CZ}(X \otimes I)\,|\Psi\rangle = [\mathrm{CZ}(X \otimes I)\mathrm{CZ}] \cdot \mathrm{CZ}\,|\Psi\rangle\,.$$

    - $[\mathrm{CZ}(X \otimes I)\mathrm{CZ}]$ is a single, unified operator acting on the ideal state.
    - Because $\mathrm{CZ}(X \otimes I)\mathrm{CZ} = X \otimes Z$, a Z error propagates down to Data 2.
  - The fault remains on the ancilla during $\mathrm{CZ}_{A\to 3}$, propagating a second Z error down to Data 3 and so on.
    - If the code only guarantees protection against weight-1 errors, this circuit destroys the fault-tolerance.

The Remedy: Introducing Shor Cat States

- To prevent a single ancilla from acting as an error-cloning vector, we must completely isolate the interaction paths.
  - Definition: A *Shor state* (aka. the cat state or the GHZ state) is a maximally entangled multi-qubit state:

$$|\text{Cat}\rangle = \frac{1}{\sqrt{2}}(|0000\rangle + |1111\rangle)$$

  - The Layout Changes: Instead of one ancilla traveling to four data qubits, we prepare a 4-qubit Cat State offline.
  - Quarantine Enforcement: Each individual qubit of the Cat State interacts with exactly one matching physical data qubit in the code block.
    - If a hardware fault occurs on any single qubit within the Cat State, it can only propagate down its own vertical track into its assigned data qubit.
    - The error weight on the data block is strictly bounded to $\text{wt}(E) \leq 1$.

The Principle of Code Concatenation

- To drive logical error rates arbitrarily low without inventing infinitely large codes, one approach is to employ a recursive, hierarchical encoding strategy called **concatenation**.
  - ◇ Level 0 (C_0): Raw physical qubits and physical gates.
    - ○ The component failure rate per clock cycle is p.
  - ◇ Level 1 (C_1): A base stabilizer code layout where n physical qubits encode 1 logical qubit (e.g., the 7-qubit Steane code).
  - ◇ Level 2 (C_2): Begin to nest the code.
    - ○ Every single physical qubit in the C_1 layout is replaced by an entire C_1 logical block.
    - ○ The system now tracks n^2 physical qubits.
  - ◇ Level k (C_k): The recursion continues to depth k, mapping 1 logical qubit to n^k physical qubits.

Tracking Error Suppression: Two Levels of Encoding

- Assume
  - ◇ The base code corrects any single fault ($t = 1$).
  - ◇ A logical block fails if two or more independent physical faults occur within the same cycle.
- Level 1 effective error (p_1): The uncorrected failure probability is dominated by the lowest-order failure event (two simultaneous faults):

$$p_1 \approx \binom{N_{\text{locations}}}{2} p^2 (1-p)^{N_{\text{locations}}-2} = cp^2.$$

- Level 2 effective error (p_2):
  - ◇ At Level 2, the circuit topology is identical, but the components are now Level 1 blocks.
  - ◇ The system fails if two or more Level 1 blocks fail:

$$p_2 = c(p_1)^2 = c^3 p^4.$$

    - ○ Misleading equation, indicating only the leading term.

The Asymptotic Limit: The Quantum Threshold Theorem

- Extending this recursive relation to an arbitrary concatenation depth k.
  - The effective logical error rate behaves as a discrete dynamical map:

$$p_k = c(p_{k-1})^2 \implies p_k = \frac{1}{c}(cp)^{2^k}. \tag{3}$$

- A strict phase transition governed by the fixed point of the map:
  - If $cp > 1$, then $p_k \longrightarrow 1$ as $k \to \infty$ (errors explode macroscopically).
    - p_k represents a probability, which is strictly capped.
  - If $cp < 1$, then $p_k \longrightarrow 0$ as $k \to \infty$ (errors vanish double-exponentially).

The Threshold Theorem

An ideal quantum circuit can be simulated using noisy physical components to an arbitrarily small target error ϵ, provided the physical component error rate p is strictly below the constant threshold value:

$$p < p_{\text{th}} = \frac{1}{c}$$

Resource Scaling and Polylogarithmic Overhead

- The tradeoff:
  - ◇ The logical error rate drops double-exponentially.
  - ◇ The physical qubit and gate overhead scales exponentially with the depth k as $O(n^k)$.
- Quantifying the overhead cost
  - ◇ To simulate a circuit containing a total of $p(n)$ gates with an overall failure probability bounded by ϵ, each logical gate requires a target accuracy of $p_k \leq \epsilon / p(n)$.
    - ○ Solving $\frac{1}{c}(cp)^{2^k} \leq \frac{\epsilon}{p(n)}$ for the required depth k yields:

$$k \sim O\left(\log\log \frac{p(n)}{\epsilon}\right)$$

    - ○ Substituting this depth back into our physical size scaling function n^k to obtain a polynomial-of-logs layout for hardware cost:

$$\text{Total Gate Complexity} = O\left(\text{poly}\left(\log \frac{p(n)}{\epsilon}\right) p(n)\right)$$

- Scalable fault-tolerance is viable because suppressing physical noise does not require an exponential explosion in hardware size.

Summary of This Lecture

- **Three obstacles** (no-cloning, continuous errors, measurement collapse) are overcome by *syndrome measurement*: ± 1 eigenvalues identify errors without touching (α, β).
- **Basic codes:** Bit-flip (corrects X); phase-flip (H-conjugated, corrects Z); Shor 9-qubit code (concatenation, corrects arbitrary single-qubit errors).
- **Knill–Laflamme (Thm. 10.1):** $PE_i^\dagger E_j P = \alpha_{ij} P$ (α Hermitian PSD) is necessary and sufficient; error subspaces are orthogonal isometric copies of the code space.
- **Discretization (Thm. 10.2):** Any $F_j = \sum_i m_{ji} E_i$ in the span of correctable $\{E_i\}$ is automatically corrected; syndrome collapse converts continuous noise to discrete Paulis.
- **CSS & stabilizer codes:** CSS nests $C_2 \subset C_1$; $k = n - \ell$ logical qubits; $H\Omega H^T = 0$. The $[[5, 1, 3]]$ code saturates $(1 + 15) \times 2 = 32 = 2^5$.
- **Clifford group:** Clifford circuits are classically simulable (Gottesman–Knill, $\mathcal{O}(n^2)$); non-Clifford T gate supplied via magic state injection $|T\rangle = (|0\rangle + e^{i\pi/4}|1\rangle)/\sqrt{2}$.
- **Threshold theorem:** Concatenation drives $p_k = (cp)^{2^k}/c \to 0$ for $p < 1/c$ with polylogarithmic gate overhead — noise is no fundamental barrier to large-scale quantum computing.