

# MASTER TEACHER'S GUIDE

## Week 5: Multi-Qubit Circuits & Noise (The Fault-Tolerant Shield)

This module addresses the foundational problem of noise on quantum hardware. Following the No-Cloning Theorem demonstration and the exploration of multi-qubit communication protocols, this unit transitions from using entanglement as a resource to using it as a protective shield against environmental decoherence.

### 1. Curriculum Overview:

Field	Detail
Target Audience	Tier 4 - Advanced Level
Design Principle	Mathematical mastery of the <b>Stabilizer Formalism</b> and computational implementation using the <b>Qiskit SDK</b> .
Learning Progression	<ol style="list-style-type: none"><li>1. <b>Formal Derivation:</b><ul style="list-style-type: none"><li>○ Students begin with the mathematical derivation of the <b>Stabilizer Formalism</b> and <b>Syndrome Derivation</b>.</li></ul></li></ol>

	<ul style="list-style-type: none"> <li>○ This involves mapping each possible single X (bit-flip) error to its unique syndrome result.</li> </ul> <p><b>2. Advanced Circuit Implementation:</b></p> <ul style="list-style-type: none"> <li>○ Students transition to using <b>Conditional Logic</b> within the Qiskit SDK.</li> <li>○ They construct a full <b>5-qubit circuit</b> (consisting of 3 data qubits and 2 ancilla qubits) to implement the 3-qubit bit-flip code.</li> <li>○ This step includes automating the process so the circuit can detect and fix errors independently.</li> </ul> <p><b>3. Protocol Analysis:</b></p> <ul style="list-style-type: none"> <li>○ The final stage involves evaluating the <b>Code Distance</b> and identifying <b>Failure Modes</b>.</li> <li>○ Students contextualize these noise mitigation strategies as the theoretical foundation for <b>Week 8 Cryptography (BB84)</b>.</li> </ul>
<b>Duration</b>	1 Week (approx. 4 x 60-90 minute sessions)
<b>Teacher Guidance</b>	Computational and Mathematical Mastery utilizing the Qiskit SDK and Tensor Product notation.

## 2. Pedagogical Framework: The Quantum Vault

Students explore the conflict between the **No-Cloning Theorem** and **Fault Tolerance**. Since quantum states cannot be copied, ancilla qubits are used to extract error syndromes without collapsing the logical data.

Focus Area	Objective (The student will be able to...)	Bloom's Level
Science/Literacy	Define the conflict between the <b>No-Cloning Theorem</b> and fault tolerance. Explain how ancilla qubits extract syndromes without collapsing data.	Understanding, Analyzing
Mathematics	Derive the <b>Stabilizer Formalism</b> for the 3-qubit bit-flip code ( $S_1=Z_0 Z_1$ , $S_2=Z_1 Z_2$ ).	Applying, Evaluating
Computational	Construct a <b>5-qubit QEC circuit</b> in Qiskit using <code>qc.measure().c_if</code> for autonomous correction.	Creating, Analyzing

### 3. Tier 4 Curriculum Sequence (4 Weeks)

The curriculum gradually builds complexity from reading comprehension to multi-qubit logic.

Module	Weeks	Core Activity	Key Quantum Concept
1. Foundational Literacy	Week 1	Comprehension Worksheets (The Magical Scroll / The Broken Printer).	<b>Unitary Evolution (Reversibility),</b> <b>The requirement of an arbitrary unknown state.</b>

<b>2. Applied Lab 1</b>	<b>Week 2</b>	The CNOT Copier Success (Composer Lab).	Confirmation that the CNOT ("Tandem Link") <i>can</i> copy classical states
<b>3. Applied Lab 2</b>	<b>Week 3</b>	The CNOT Cloning Failure (Composer Lab).	<b>No-Cloning Theorem</b> <b>Demonstration.</b> CNOT on a superposition state creates <b>Entanglement</b> (a Bell state) instead of two independent copies.
<b>4. Final Logic Project</b>	<b>Week 4</b>	The Resource Swap (Comparative Qiskit Simulation).	Protocol Efficiency and the Causal Requirement of the Classical Channel.
<b>5. Error Mitigation</b>	<b>Week 5</b>	<b>The Fault-Tolerant Shield</b> (5-qubit Qiskit Simulation).	<b>Stabilizer Formalism, Syndrome Measurement, and the No-Cloning Theorem</b> Constraint on Hardware Fidelity.

## 4. Qiskit Lab: The Fault-Tolerant Shield (3-Qubit Bit-Flip Code)

**Objective:** Construct a 5-qubit QEC circuit using `qc.measure().c_if` for autonomous correction.

1. Encoding: Prepare the protected state across three data qubits (`q_0`, `q_1`, `q_2`).
2. Error Simulation: Inject a single bit-flip error (`X` gate) to test the code's effectiveness.
3. Syndrome Measurement: Use two ancilla qubits (`a_0`, `a_1`) to measure the parity of data qubit pairs.
4. Autonomous Correction: Apply `qc.measure().c_if` to automatically trigger a corrective `X` gate based on the syndrome results.
5. Verification: Use the Statevector Simulator to confirm the final state matches the initial input, proving the "healing" was successful.

### Conclusion and Next Steps

## 5. Next Steps

This module serves as the theoretical underpinning for **Week 8: BB84 Protocol**. Students will contrast internal error correction (QEC) with external adversarial detection (QKD).

# Week 5 Lesson Plan: The Fault-Tolerant Shield

## 1. Module Overview

- **Unit Title:** Multi-Qubit Circuits & Noise.
- **Topic:** Quantum Error Correction (QEC) via the 3-Qubit Bit-Flip Code.
- **Objective:** Students will construct a 5-qubit QEC circuit in the Qiskit SDK to detect and autonomously correct a single bit-flip error.

## 2. Learning Objectives

By the end of this lesson, students will be able to:

- **Define** the conflict between the No-Cloning Theorem and the need for fault tolerance.
- **Derive** the Stabilizer Formalism for the 3-qubit bit-flip code ( $S_1=Z_0Z_1$ ,  $S_2=Z_1Z_2$ ).
- **Implement** a syndrome measurement block using ancilla qubits.
- **Execute** autonomous correction using Qiskit's conditional logic (`qc.measure().c_if`).

## 3. Required Resources

- **IBM Quantum Lab:** Jupyter Notebooks for Python-based Qiskit SDK development.
- **Qiskit SDK:** Specifically utilizing the `c_if` conditional feature.
- **Simulator:** Qiskit Statevector Simulator for result verification.

---

## 4. Step-by-Step Instructions

### Part 1: Encoding the Logical Qubit

1. **Initialize Data Qubits:** Create a quantum circuit with 3 data qubits (`q_0`, `q_1`, `q_2`) and 2 ancilla qubits (`a_0`, `a_1`).
2. **Encode State:** Prepare the logical state. To encode an arbitrary state  $\psi$ , apply CNOT gates from `q_0` to `q_1` and `q_2`.

### Part 2: Error Simulation

1. **Inject Noise:** Manually apply a single X gate (bit-flip) to one of the data qubits (e.g., `qc.x(q_1)`) to simulate environmental decoherence.
2. **Constraint:** Remind students that the code only protects against a **single** bit-flip in this configuration.

### Part 3: Syndrome Extraction (The Diagnostic)

1. **Measure Parity:** Use the two ancilla qubits to measure the stabilizers:
  - **Ancilla a\_0:** Measures  $Z_0Z_1$  (checks if  $q_0$  and  $q_1$  are the same).
  - **Ancilla a\_1:** Measures  $Z_1Z_2$  (checks if  $q_1$  and  $q_2$  are the same).
2. **Classical Storage:** Map these measurements to a classical register to create a 2-bit "syndrome" string.

### Part 4: Autonomous Correction

1. **Conditional Logic:** Use the `.c_if()` method to apply a corrective X gate based on the syndrome:
  - If syndrome is **01**: Flip  $q_0$ .
  - If syndrome is **11**: Flip  $q_1$ .
  - If syndrome is **10**: Flip  $q_2$ .
  - If syndrome is **00**: No error detected.

### Part 5: Verification

1. **Simulation:** Run the circuit using the `Statevector` simulator.
2. **Comparison:** Compare the final statevector against the initial input state to confirm the "healing" was successful and the data qubit was preserved.

## 5. Assessment & Discussion

- **Discussion Point:** Why can't we just measure the data qubits to see if they flipped? (Answer: Measurement collapses the superposition, violating the No-Cloning constraint).
- **Technical Write-up:** Students must analyze the "qubit overhead"—the fact that 5 physical qubits are required to protect 1 logical qubit.
- **Next Steps:** Contrast this local error correction with the external eavesdropper detection found in the **BB84 Protocol** (Week 8).