tiny-qpu

Technical Manual & Quantum Computing Primer

Comprehensive Reference

Quantum computing foundations, gate mathematics, algorithm deep dives, simulator architecture, worked examples, and 40+ exercises.

From first qubit to Shor's algorithm.

Version 2.0 — February 2026

github.com/SKBiswas1998/tiny-qpu

Contents

Chapter 1

Mathematical Foundations of Quantum Computing

1.1 Complex Numbers

Quantum mechanics is built on complex numbers. A complex number $z = a + bi$ has real part a and imaginary part b, where $i^2 = -1$. The **complex conjugate** is $z^* = a - bi$. The **modulus** is:

$$|z| = \sqrt{a^2 + b^2}, \qquad |z|^2 = zz^* = a^2 + b^2 \tag{1.1}$$

The **polar form** connects magnitude and phase:

$$z = |z|\, e^{i\theta} = |z|(\cos\theta + i\sin\theta), \qquad \theta = \mathrm{atan2}(b,\, a) \tag{1.2}$$

Two amplitudes with opposite phases cancel (*destructive interference*); two with identical phases reinforce (*constructive interference*). Every quantum algorithm exploits interference.

Euler's formula and important special cases:

$$e^{i\pi} = -1, \qquad e^{i\pi/2} = i, \qquad e^{i\pi/4} = \frac{1+i}{\sqrt{2}} \tag{1.3}$$

1.2 Linear Algebra Essentials

Quantum states live in **Hilbert space** — a complex vector space equipped with an inner product. For n qubits the state space has dimension 2^n.

1.2.1 Inner Product

$$\langle \mathbf{a} | \mathbf{b} \rangle = \sum_i a_i^* b_i \tag{1.4}$$

Orthogonal states satisfy $\langle a|b\rangle = 0$. An orthonormal basis satisfies $\langle i|j\rangle = \delta_{ij}$.

1.2.2 Tensor (Kronecker) Product

Combining system A (dimension d_A) with B (dimension d_B) yields dimension $d_A \times d_B$. For two qubits: $2 \times 2 = 4$ dimensional. Explicitly:

$$\begin{pmatrix} a_1 \\ a_2 \end{pmatrix} \otimes \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} = \begin{pmatrix} a_1 b_1 \\ a_1 b_2 \\ a_2 b_1 \\ a_2 b_2 \end{pmatrix} \tag{1.5}$$

1.2.3 Unitary Matrices

A matrix U is **unitary** if $UU^\dagger = U^\dagger U = I$, where $U^\dagger$ is the conjugate transpose. All quantum gates are unitary, guaranteeing:

- Reversibility: $U^{-1} = U^\dagger$.
- Norm preservation: $\|U\,|\psi\rangle\,\| = \|\,|\psi\rangle\,\|$ (total probability conserved).

1.2.4 Hermitian Matrices

A matrix H is **Hermitian** if $H = H^\dagger$. Observables are Hermitian operators: eigenvalues are real, eigenvectors form an orthonormal basis. The Pauli matrices X, Y, Z are both unitary *and* Hermitian.

1.3 Dirac (Bra-Ket) Notation

Notation	Meaning	Example
$\|\psi\rangle$	Column vector (ket)	$\|0\rangle = (1,\ 0)^T$
$\langle\psi\|$	Row vector (bra)	$\langle 0\| = (1,\ 0)$
$\langle\phi\|\psi\rangle$	Inner product (scalar)	$\langle 0\|1\rangle = 0$
$\|\psi\rangle\langle\phi\|$	Outer product (operator)	$\|0\rangle\langle 0\| = \mathrm{diag}(1,0)$
$U\,\|\psi\rangle$	Operator acting on state	$X\,\|0\rangle = \|1\rangle$
$\langle\psi\|A\|\phi\rangle$	Matrix element	$\langle 0\|Z\|0\rangle = 1$

Table 1.1: Dirac notation summary.

Exercises — Chapter 1

1.1. Compute $|z|^2$ for $z = (1+i)/\sqrt{2}$. Convert to polar form and find the phase angle θ.

1.2. Verify that $H = \frac{1}{\sqrt{2}}\left(\begin{smallmatrix} 1 & 1 \\ 1 & -1 \end{smallmatrix}\right)$ is unitary by computing $HH^\dagger$ and showing it equals I.

1.3. Compute $|+\rangle \otimes |0\rangle$ where $|+\rangle = (|0\rangle + |1\rangle)/\sqrt{2}$. Write as a 4-component vector. Which basis states have nonzero amplitude?

1.4. Show that the eigenvalues of Pauli-Z are $+1$ and -1, with eigenvectors $|0\rangle$ and $|1\rangle$. *Hint: solve* $Z\,|v\rangle = \lambda\,|v\rangle$.

Chapter 2

The Qubit

2.1 Computational Basis States

A **qubit** is the fundamental unit of quantum information. Unlike a classical bit, a qubit exists in a **superposition**:

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle \tag{2.1}$$

where $\alpha, \beta \in \mathbb{C}$ are **probability amplitudes** satisfying:

$$|\alpha|^2 + |\beta|^2 = 1 \tag{2.2}$$

The computational basis states are:

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \qquad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \tag{2.3}$$

Upon measurement, the qubit **collapses** to $|0\rangle$ with probability $P(0) = |\alpha|^2$ or to $|1\rangle$ with probability $P(1) = |\beta|^2$. The superposition is irreversibly destroyed.

2.2 The Bloch Sphere

Any single-qubit pure state can be parameterized as:

$$|\psi\rangle = \cos\frac{\theta}{2}\,|0\rangle + e^{i\phi}\sin\frac{\theta}{2}\,|1\rangle \tag{2.4}$$

where $\theta \in [0, \pi]$ and $\phi \in [0, 2\pi)$. The Bloch vector coordinates are:

$$x = \sin\theta\cos\phi, \qquad y = \sin\theta\sin\phi, \qquad z = \cos\theta \tag{2.5}$$

State	θ	ϕ	**Bloch** (x, y, z)	**Meaning**
$\|0\rangle$	0	—	$(0, 0, +1)$	North pole
$\|1\rangle$	π	—	$(0, 0, -1)$	South pole
$\|+\rangle$	$\pi/2$	0	$(+1, 0, 0)$	Positive x
$\|-\rangle$	$\pi/2$	π	$(-1, 0, 0)$	Negative x
$\|+i\rangle$	$\pi/2$	$\pi/2$	$(0, +1, 0)$	Positive y
$\|-i\rangle$	$\pi/2$	$3\pi/2$	$(0, -1, 0)$	Negative y

Table 2.1: Standard states on the Bloch sphere.

Quantum gates correspond to **rotations** of the Bloch sphere.

Note

In the Quantum Lab, the Bloch sphere displays the state vector in real time. When a qubit is entangled, its reduced state is mixed and the Bloch vector shrinks toward the origin — a visual signature of entanglement.

2.3 Global Phase vs. Relative Phase

A **global phase** $e^{i\gamma}\left|\psi\right\rangle$ is physically undetectable. A **relative phase** between components is observable:

$$\left|+\right\rangle = \tfrac{1}{\sqrt{2}}(\left|0\right\rangle + \left|1\right\rangle), \qquad \left|-\right\rangle = \tfrac{1}{\sqrt{2}}(\left|0\right\rangle - \left|1\right\rangle) \tag{2.6}$$

After applying H: $H\left|+\right\rangle = \left|0\right\rangle$ (always 0), $H\left|-\right\rangle = \left|1\right\rangle$ (always 1). Relative phase is the resource quantum algorithms manipulate.

2.4 Mixed States and the Density Matrix

A pure state has density matrix $\rho = |\psi\rangle\langle\psi|$. A **mixed state** is:

$$\rho = \sum_i p_i \left|\psi_i\right\rangle\!\left\langle\psi_i\right|, \qquad \sum_i p_i = 1 \tag{2.7}$$

Purity: $\mathrm{Tr}(\rho^2) = 1$ for pure states, < 1 for mixed. Entangled qubits have mixed reduced density matrices.

Exercises — Chapter 2

2.1. Given $\left|\psi\right\rangle = \frac{1}{2}\left|0\right\rangle + \frac{\sqrt{3}}{2}\left|1\right\rangle$, compute $P(0)$, $P(1)$, and find Bloch angles θ, ϕ.

2.2. For $\left|\psi\right\rangle = \frac{1}{\sqrt{2}}\left|0\right\rangle + \frac{i}{\sqrt{2}}\left|1\right\rangle$, identify θ and ϕ. Verify: apply $R_y(\pi/2)$ then S to $\left|0\right\rangle$ in the Lab.

2.3. Apply H to $\left|0\right\rangle$ in step mode. Record the Bloch vector. Add Z after H. How does it change?

2.4. Show $\left|+\right\rangle$ and $-\left|+\right\rangle$ are physically identical by computing the density matrix for each.

Chapter 3

Quantum Gates: Complete Reference

All quantum gates are **unitary operators**. Gate U applied to $|\psi\rangle$ gives $U|\psi\rangle$; the inverse is $U^\dagger$.

3.1 Pauli Gates

The Pauli matrices with I form a basis for all 2×2 Hermitian matrices:

$$X^2 = Y^2 = Z^2 = I, \qquad XY = iZ \text{ (cyclic)} \tag{3.1}$$

3.1.1 Pauli-X (Bit Flip)

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \tag{3.2}$$

$X|0\rangle = |1\rangle$, $X|1\rangle = |0\rangle$. Bloch: 180ř rotation about x-axis. Eigenvalues: ±1 with eigenvectors $|\pm\rangle$.

3.1.2 Pauli-Y (Bit + Phase Flip)

$$Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \tag{3.3}$$

$Y|0\rangle = i|1\rangle$, $Y|1\rangle = -i|0\rangle$. Relation: $Y = iXZ$. Bloch: 180ř about y-axis.

3.1.3 Pauli-Z (Phase Flip)

$$Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \tag{3.4}$$

$Z|0\rangle = |0\rangle$, $Z|1\rangle = -|1\rangle$. Adds phase π to $|1\rangle$. Bloch: 180ř about z-axis. Note: $Z|+\rangle = |-\rangle$.

3.2 The Hadamard Gate

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \tag{3.5}$$

$H|0\rangle = |+\rangle$, $H|1\rangle = |-\rangle$. Self-inverse: $H^2 = I$. Key identities:

$$HZH = X, \qquad HXH = Z \tag{3.6}$$

Quantum parallelism:

$$H^{\otimes n} |0\rangle^{\otimes n} = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle \tag{3.7}$$

3.3 Phase Gates: S, T, $P(\theta)$

Gate	Matrix	Phase on $\|1\rangle$	Bloch	Relation
S	$\left(\begin{smallmatrix} 1 & 0 \\ 0 & i \end{smallmatrix}\right)$	$\pi/2$	z by 90ř	$S^2 = Z$
T	$\begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix}$	$\pi/4$	z by 45ř	$T^2 = S$
$P(\theta)$	$\begin{pmatrix} 1 & 0 \\ 0 & e^{i\theta} \end{pmatrix}$	θ	z by θ	General phase

Table 3.1: Phase gates.

Important

The gate set $\{H, T, \text{CNOT}\}$ is **universal** — any unitary can be approximated to arbitrary precision (Solovay–Kitaev theorem).

3.4 Rotation Gates

$$R_x(\theta) = \cos\tfrac{\theta}{2}\, I - i\sin\tfrac{\theta}{2}\, X = \begin{pmatrix} \cos\frac{\theta}{2} & -i\sin\frac{\theta}{2} \\ -i\sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{pmatrix} \tag{3.8}$$

$$R_y(\theta) = \cos\tfrac{\theta}{2}\, I - i\sin\tfrac{\theta}{2}\, Y = \begin{pmatrix} \cos\frac{\theta}{2} & -\sin\frac{\theta}{2} \\ \sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{pmatrix} \tag{3.9}$$

$$R_z(\theta) = \begin{pmatrix} e^{-i\theta/2} & 0 \\ 0 & e^{i\theta/2} \end{pmatrix} \tag{3.10}$$

ZYZ **decomposition:** Any single-qubit unitary:

$$U = e^{i\alpha}\, R_z(\beta)\, R_y(\gamma)\, R_z(\delta) \tag{3.11}$$

3.5 Multi-Qubit Gates

3.5.1 CNOT (CX)

$$\text{CNOT} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} = |0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes X \tag{3.12}$$

Flips target iff control is $|1\rangle$. Primary **entangling gate**:

$$|00\rangle \xrightarrow{H\otimes I} \frac{|00\rangle + |10\rangle}{\sqrt{2}} \xrightarrow{\text{CNOT}} \frac{|00\rangle + |11\rangle}{\sqrt{2}} = \left|\Phi^+\right\rangle \tag{3.13}$$

3.5.2 CZ (Controlled-Z)

$|11\rangle \to -|11\rangle$, all others unchanged. CZ is symmetric: $\mathrm{CZ}(a, b) = \mathrm{CZ}(b, a)$.

3.5.3 SWAP

$\mathrm{SWAP}\,|a, b\rangle = |b, a\rangle$. Decomposition:

$$\mathrm{SWAP} = \mathrm{CNOT}(0, 1)\ \mathrm{CNOT}(1, 0)\ \mathrm{CNOT}(0, 1) \tag{3.14}$$

3.5.4 Toffoli (CCX)

Flips target iff *both* controls are $|1\rangle$. **Universal for classical computation.** Used in Grover's oracle and arithmetic circuits.

3.5.5 Fredkin (CSWAP)

Controlled-SWAP. Used in quantum comparison and the SWAP test.

3.5.6 $CR_z(\theta)$

Controlled-R_z. Essential in QFT circuits where controlled rotations by $2\pi/2^k$ create Fourier phases.

Exercises — Chapter 3

3.1. Verify $HZH = X$ by matrix multiplication. What does this tell you about how H relates the Z-basis and X-basis?

3.2. Compute $R_x(\pi/2)\,|0\rangle$ by hand. Express in Bloch coordinates. Verify in the Lab.

3.3. Build H, T, H in the Lab. Record probabilities. Replace T with S and compare.

3.4. Prove CNOT creates entanglement: show $(|00\rangle + |11\rangle)/\sqrt{2}$ cannot be factored as $|a\rangle \otimes |b\rangle$.

3.5. Decompose SWAP into three CNOTs by computing on all four basis states.

Chapter 4

Multi-Qubit Systems and Entanglement

4.1 Tensor Products and State Spaces

An n-qubit system has state space dimension 2^n. The general state:

$$|\psi\rangle = \sum_{x=0}^{2^n-1} c_x |x\rangle , \qquad \sum_x |c_x|^2 = 1 \tag{4.1}$$

For 2 qubits, the basis is $\{|00\rangle, |01\rangle, |10\rangle, |11\rangle\}$ requiring 4 complex amplitudes. For 30 qubits: over 10^9 amplitudes — this exponential scaling makes classical simulation intractable.

A **separable state** can be written as $|\psi\rangle = |a\rangle \otimes |b\rangle$. An **entangled state** cannot.

4.2 The Four Bell States

The Bell states form a maximally entangled orthonormal basis for two qubits:

$$\left|\Phi^+\right\rangle = \frac{|00\rangle + |11\rangle}{\sqrt{2}} \qquad \text{Same outcomes, no phase} \tag{4.2}$$

$$|\Phi^-\rangle = \frac{|00\rangle - |11\rangle}{\sqrt{2}} \qquad \text{Same outcomes, } \pi \text{ phase} \tag{4.3}$$

$$\left|\Psi^+\right\rangle = \frac{|01\rangle + |10\rangle}{\sqrt{2}} \qquad \text{Opposite outcomes, no phase} \tag{4.4}$$

$$|\Psi^-\rangle = \frac{|01\rangle - |10\rangle}{\sqrt{2}} \qquad \text{Opposite outcomes, } \pi \text{ phase} \tag{4.5}$$

State	**Circuit**	P(**same**)	P(**diff**)
$\|\Phi^+\rangle$	$H(q_0)$, CX(q_0, q_1)	1	0
$\|\Phi^-\rangle$	$X(q_0)$, $H(q_0)$, CX(q_0, q_1)	1	0
$\|\Psi^+\rangle$	$H(q_0)$, CX(q_0, q_1), $X(q_1)$	0	1
$\|\Psi^-\rangle$	$X(q_0)$, $H(q_0)$, CX(q_0, q_1), $X(q_1)$	0	1

Table 4.1: Bell state preparation circuits and measurement correlations.

Measuring one qubit of a Bell pair **instantaneously** determines the other's outcome. This cannot transmit information faster than light (the result is random).

4.3 GHZ and W States

$$|\mathrm{GHZ}\rangle = \frac{|000\rangle + |111\rangle}{\sqrt{2}} \tag{4.6}$$

$$|W\rangle = \frac{|001\rangle + |010\rangle + |100\rangle}{\sqrt{3}} \tag{4.7}$$

GHZ: maximally entangled but fragile — losing one qubit destroys all entanglement. W: more robust — tracing out one qubit still leaves the other two entangled. These belong to different **entanglement classes**.

4.4 Quantifying Entanglement

For a bipartite state $|\psi\rangle_{AB}$, the **von Neumann entropy**:

$$S(\rho_A) = -\mathrm{Tr}(\rho_A \log_2 \rho_A) \tag{4.8}$$

where $\rho_A = \mathrm{Tr}_B(|\psi\rangle\langle\psi|)$. $S = 0$ for separable states, $S = 1$ for maximally entangled qubit pairs.

Exercises — Chapter 4

4.1. Build all four Bell states in the Lab. Record the probability distribution and Bloch vector length for each qubit.

4.2. Create $(|00\rangle + |01\rangle + |10\rangle + |11\rangle)/2$ using $H(q_0), H(q_1)$. Is it entangled? Write it as a tensor product to prove your answer.

4.3. Build a 4-qubit GHZ state. How many basis states have nonzero probability?

4.4. Build the W state on 3 qubits. *Hint:* $R_y(2\arccos\left(1/\sqrt{3}\right))$ *on* q_0*, then conditional gates.* Compare Bloch vectors to GHZ.

Chapter 5

Measurement Theory

5.1 Born's Rule (Projective Measurement)

Measuring observable O with eigenvalues $\{\lambda_k\}$ and projectors $\{P_k = |k\rangle\langle k|\}$:

$$P(\text{outcome } k) = \langle\psi|P_k|\psi\rangle = |\langle k|\psi\rangle|^2 \tag{5.1}$$

$$\text{Post-measurement state:} \quad |\psi'\rangle = \frac{P_k|\psi\rangle}{\sqrt{P(k)}} \tag{5.2}$$

Measurement is **irreversible**, **probabilistic**, and **disturbing**. It collapses superposition and can create or destroy entanglement.

5.2 Measurement Bases

The computational (Z) basis $\{|0\rangle, |1\rangle\}$ is the default. To measure in a different basis, apply a change-of-basis unitary before measurement:

Basis	Pre-gate	Result 0	Result 1
X: $\{\|+\rangle, \|-\rangle\}$	H	Was $\|+\rangle$	Was $\|-\rangle$
Y: $\{\|+i\rangle, \|-i\rangle\}$	$S^\dagger$ then H	Was $\|+i\rangle$	Was $\|-i\rangle$
Arbitrary axis $\hat{n}$	Appropriate rotation	Spin-up	Spin-down

Table 5.1: Measurement in different bases.

5.3 Statevector vs. Shot-Based Results

`tiny-qpu` provides two complementary views:

Exact probabilities (statevector): $P(k) = |c_k|^2$. Theoretical values with infinite precision.

Sampled histogram (shots): Each shot randomly collapses according to exact probabilities. With N shots, the count for state k follows Binomial$(N, P(k))$ with standard deviation $\sqrt{NP(k)(1-P(k))}$. More shots = less noise.

Note

A real quantum computer only gives the histogram. The simulator's ability to show exact probabilities alongside statistical counts is a powerful learning tool.

5.4 Partial Measurement

Measuring a subset of qubits projects onto the measured subspace. If we measure the first qubit of $\alpha\,|00\rangle + \beta\,|01\rangle + \gamma\,|10\rangle + \delta\,|11\rangle$ and get $|0\rangle$, the post-measurement state is:

$$|\psi'\rangle = \frac{\alpha\,|00\rangle + \beta\,|01\rangle}{\sqrt{|\alpha|^2 + |\beta|^2}} \tag{5.3}$$

Exercises — Chapter 5

5.1. Create $R_y(\pi/3)\,|0\rangle$. Compute exact probabilities. Run with 100 and 10,000 shots. Calculate expected standard deviation and compare.

5.2. Create $|+\rangle$ (H on $|0\rangle$). Measure: expect 50/50. Now add H before measurement ($HH = I$). What happens? Explain.

5.3. Create Bell state $|\Phi^+\rangle$. What are the phases of the nonzero amplitudes? How would $|\Phi^-\rangle$ differ?

Chapter 6

The tiny-qpu Simulator Engine

6.1 Architecture Overview

`tiny-qpu` is a classical simulator that maintains the full quantum state in memory and applies gate operations as linear algebra transformations.

Component	Module	Purpose
Circuit builder	`circuit.py`	Gate sequence construction, QASM parsing
Gate library	`gates.py`	35+ gate matrices (NumPy arrays)
Statevector backend	`backends/statevector.py`	Pure-state simulation
Density matrix backend	`backends/density_matrix.py`	Mixed states, noise channels
QASM parser	`qasm/parser.py`	OpenQASM 2.0 tokenizer
Dashboard	`dashboard/server.py`	Flask REST API for the Lab UI

Table 6.1: Simulator architecture.

6.2 Statevector Backend

Stores the quantum state as a complex vector of 2^n amplitudes. A single-qubit gate U on qubit k is applied via the tensor product $I^{\otimes(k)} \otimes U \otimes I^{\otimes(n-k-1)}$.

Qubits	Amplitudes	Memory	Time
1–6	2–64	$<$ 1 KB	Instant
10	1,024	16 KB	$<$ 1 ms
15	32,768	512 KB	$\sim$10 ms
20	1,048,576	16 MB	$\sim$1 s
25	33,554,432	512 MB	$\sim$minutes

Table 6.2: Statevector backend scaling.

6.3 Density Matrix Backend

Stores ρ as a $2^n \times 2^n$ complex matrix. Gate application: $\rho' = U\rho U^\dagger$. Noise via Kraus operators: $\rho' = \sum_k E_k \rho E_k^\dagger$. Memory: 16×4^n bytes. Practical limit: 12–15 qubits.

6.4 Python API

```
from tiny_qpu.circuit import QuantumCircuit

qc = QuantumCircuit(2)
qc.h(0)                 # Hadamard on qubit 0
qc.cx(0, 1)             # CNOT: control=0, target=1
result = qc.simulate(shots=1024)

print(result.counts)         # {'00': ~512, '11': ~512}
print(result.statevector)    # [0.707+0j, 0, 0, 0.707+0j]
print(result.probabilities) # [0.5, 0.0, 0.0, 0.5]
```

```
# OpenQASM integration
from tiny_qpu.qasm import parse_qasm

qasm = '''OPENQASM 2.0;
include "qelib1.inc";
qreg q[2];
h q[0];
cx q[0],q[1];'''
circuit = parse_qasm(qasm)
result = circuit.simulate(shots=1000)
```

Exercises — Chapter 6

6.1. Using the Python API, create a 3-qubit GHZ state and print the statevector. Verify exactly two nonzero amplitudes, each $1/\sqrt{2}$.

6.2. Write a QASM circuit for the 2-qubit Deutsch–Jozsa algorithm. Execute with `parse_qasm`. What does the measurement tell you about the oracle?

Chapter 7

The Interactive Quantum Lab

7.1 Launching

The Lab runs as a native Windows desktop application (pywebview + Flask):

```
# Native window (recommended)
tiny-qpu.exe                          # standalone build
python tiny_qpu_launcher.py           # from source

# Browser fallback
python tiny_qpu_launcher.py --browser
```

7.2 Interface Layout

Header
Logo, qubit selector (1–8), shot count input.

Left Panel
Gate palette: 20 gates organized by category (single-qubit fixed, rotation, multi-qubit, measurement). Below: 8 preset circuits.

Center Panel
Circuit builder: qubit wires, placed gates, toolbar (Run, Step, Clear, Undo), status bar showing gate count and depth.

Right Panel
Bloch sphere per qubit, probability bars, measurement histogram, amplitude table (state / amplitude / phase / probability), QASM editor.

7.3 Building Circuits

Placing gates: Click a gate in the palette then click a qubit wire, or drag directly. Gates auto-place in the earliest available column.

Multi-qubit gates: Click the top qubit (control). The target is automatically assigned to adjacent qubits. CX/CZ/SWAP span 2 qubits; CCX/CSWAP span 3.

Rotation gates: A dialog prompts for the angle in radians with quick-select buttons for π, $\pi/2$, $\pi/4$, $\pi/3$, $-\pi/2$, and 0.

Removing gates: Click any placed gate to remove it, or press Ctrl+Z to undo.

7.4 Step-by-Step Mode

Press **S** or click Step. An amber indicator appears. Use arrow keys or Next/Prev buttons to advance gate by gate. At each step the right panel shows the intermediate quantum state.

What to observe: (1) Which gate is highlighted, (2) Bloch vector movement, (3) probability redistribution, (4) states appearing/disappearing in the amplitude table, (5) purity changes.

Note

Step 0 shows the initial state $|0\cdots0\rangle$ before any gates. This is the most effective way to understand how each gate transforms the quantum state.

7.5 Preset Circuits

Preset	**Concept**	**Key Observation**
Bell State	Entanglement	Bloch vectors shrink; only $|00\rangle$, $|11\rangle$
GHZ State	Multi-party entanglement	3 mixed qubits, globally pure
Superposition	H on all qubits	All 2^n states equally likely
Teleportation	State transfer	q_0's state appears on q_2
Deutsch–Jozsa	Quantum parallelism	Single query decides constant vs balanced
QFT (2-qubit)	Fourier transform	Phase info in relative phases
Grover	Amplitude amplification	Target goes from 25% to ∼100%
Bit-Flip Code	Error correction	Encoded state survives X error

Table 7.1: Preset circuits and learning objectives.

7.6 REST API

Endpoint	**Method**	**Description**
`/api/simulate`	POST	Run circuit, return full results
`/api/step`	POST	Partial execution (first k gates)
`/api/qasm/import`	POST	Parse QASM to gate list
`/api/qasm/export`	POST	Gate list to QASM string
`/api/presets`	GET	List available presets
`/api/presets/<name>`	GET	Get specific preset
`/api/gates`	GET	Gate catalog
`/health`	GET	Health check

Table 7.2: REST API endpoints.

Chapter 8

Algorithm Deep Dives

8.1 Quantum Teleportation

Transfers unknown state $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ from Alice (q_0) to Bob (q_2) using one shared Bell pair and two classical bits. The original is destroyed (no-cloning theorem).

Protocol:

1. **Prepare:** q_0 in $|\psi\rangle$. Create Bell pair: $H(q_1)$, $\text{CX}(q_1, q_2)$.
2. **Bell measurement:** $\text{CX}(q_0, q_1)$, $H(q_0)$. Measure q_0 and q_1.
3. **Correction:** If $q_1 = 1$: apply $X(q_2)$. If $q_0 = 1$: apply $Z(q_2)$.
4. **Result:** q_2 is now in state $|\psi\rangle$.

The three-qubit state before measurement expands as:

$$|\Psi\rangle = \frac{1}{2}\Big[|00\rangle(\alpha|0\rangle+\beta|1\rangle)+|01\rangle(\alpha|1\rangle+\beta|0\rangle)+|10\rangle(\alpha|0\rangle-\beta|1\rangle)+|11\rangle(\alpha|1\rangle-\beta|0\rangle)\Big] \quad (8.1)$$

Each Bell measurement outcome corresponds to a Pauli correction on q_2.

8.2 Deutsch–Jozsa Algorithm

Determines whether $f : \{0,1\}^n \to \{0,1\}$ is **constant** or **balanced** with a single query (classically requires up to $2^{n-1} + 1$).

Circuit:

1. Initialize: first n qubits in $|0\rangle$, ancilla in $|1\rangle$.
2. Apply H to all qubits.
3. Apply oracle U_f: $|x, y\rangle \to |x, y \oplus f(x)\rangle$.
4. Apply H to first n qubits.
5. Measure: all zeros ⇒ constant; anything else ⇒ balanced.

Key insight (phase kickback): With ancilla in $|-\rangle$:

$$U_f|x\rangle|-\rangle = (-1)^{f(x)}|x\rangle|-\rangle \quad (8.2)$$

The function value becomes a **phase**, which the final Hadamard converts to a measurable amplitude.

8.3 Grover's Search

Finds a marked item in an unsorted database of $N = 2^n$ items using $O(\sqrt{N})$ queries.

Algorithm:

1. **Initialize:** $H^{\otimes n}|0\rangle^{\otimes n}$ (uniform superposition).

2. **Repeat** $\approx \frac{\pi}{4}\sqrt{N}$ times:
   (a) **Oracle:** $O\left|x\right\rangle = (-1)^{\delta_{x,w}}\left|x\right\rangle$ (flip amplitude of target $\left|w\right\rangle$).
   (b) **Diffusion:** $D = 2\left|s\right\rangle\!\left\langle s\right| - I$ where $\left|s\right\rangle$ is the uniform superposition.
3. **Measure.**

Geometric picture: The state lives in the 2D plane spanned by $\left|w\right\rangle$ and $\left|w^{\perp}\right\rangle$. Each iteration rotates by $2\arcsin\left(1/\sqrt{N}\right)$ toward $\left|w\right\rangle$. For $N = 4$, one iteration reaches $\left|w\right\rangle$ exactly.

8.4 Quantum Fourier Transform (QFT)

$$\text{QFT}\left|j\right\rangle = \frac{1}{\sqrt{2^n}}\sum_{k=0}^{2^n-1} e^{2\pi ijk/2^n}\left|k\right\rangle \tag{8.3}$$

Circuit: For each qubit k (top to bottom): apply $H(k)$, then $CR_z(\pi/2^m)$ from k to qubits $k+1, k+2, \ldots$ Finally, reverse qubit order with SWAPs. Total: $O(n^2)$ gates.

Applications: Core subroutine of Shor's algorithm, quantum phase estimation, and quantum simulation.

8.5 Shor's Factoring Algorithm

Factors integer N in $O(\log^3 N)$ operations (exponentially faster than best classical).

Steps:

1. Choose random $a < N$. If $\gcd(a, N) > 1$, done.
2. **Quantum step:** Find the period r of $f(x) = a^x \bmod N$ using quantum phase estimation + QFT.
3. If r is even and $a^{r/2} \not\equiv -1 \pmod{N}$, compute $\gcd(a^{r/2} \pm 1, N)$.
4. At least one GCD is a nontrivial factor with probability $\geq 1/2$.

Example: For $N = 15$, $a = 7$: period $r = 4$, giving $\gcd(7^2 + 1, 15) = \gcd(50, 15) = 5$ and $\gcd(7^2 - 1, 15) = \gcd(48, 15) = 3$.

8.6 Variational Quantum Eigensolver (VQE)

Hybrid quantum-classical algorithm for ground state energies. Based on the **variational principle**:

$$E(\vec{\theta}) = \left\langle\psi(\vec{\theta})\middle|H\middle|\psi(\vec{\theta})\right\rangle \geq E_{\text{ground}} \tag{8.4}$$

Loop:

1. **Quantum:** Prepare ansätz $\left|\psi(\vec{\theta})\right\rangle$ with parameterized gates (R_y, CNOT).
2. **Quantum:** Measure $\langle H\rangle$ by decomposing into Pauli terms.
3. **Classical:** Optimizer (COBYLA, L-BFGS-B) updates $\vec{\theta}$ to minimize $E(\vec{\theta})$.
4. Repeat until convergence.

`tiny-qpu` supports H_2, LiH, BeH_2, H_2O with pre-computed Hamiltonians and the UCCSD ansätz.

8.7 QAOA

Solves combinatorial optimization (MaxCut, TSP, portfolio) by alternating problem and mixer Hamiltonians:

$$\left|\vec{\gamma}, \vec{\beta}\right\rangle = \prod_{p=1}^{P} e^{-i\beta_p H_M}\, e^{-i\gamma_p H_C}\, \left|+\right\rangle^{\otimes n} \tag{8.5}$$

Parameters optimized classically. Higher P improves approximation ratio.

Exercises — Chapter 8

8.1. Load the Teleportation preset. Step through and identify: Bell pair creation, Bell measurement, and correction steps.

8.2. Implement 3-qubit Grover for target $|101\rangle$. How many iterations are optimal for $N = 8$? Build and verify.

8.3. Use the Python API to factor 15 via `algorithms/shor.py`. Run 20 times. What fraction succeed?

8.4. Use VQE to compute H_2 ground state energy at bond lengths $0.5, 0.735, 1.0, 1.5, 2.0$ Å. Plot E vs distance. Where is the equilibrium?

Chapter 9

Quantum Error Correction

9.1 Types of Quantum Errors

Error	**Effect**	**Pauli**	**Cause**
Bit flip	$\|0\rangle \leftrightarrow \|1\rangle$	X	EM interference
Phase flip	$\|+\rangle \leftrightarrow \|-\rangle$	Z	Dephasing
Bit + phase	Combined	$Y = iXZ$	Multiple sources
Depolarizing	Random Pauli	$pI + \frac{1-p}{3}(X+Y+Z)$	General decoherence
Amplitude damping	Energy decay (T_1)	Non-unitary	Spontaneous emission
Dephasing	Phase randomization (T_2)	Random R_z	Environmental noise

Table 9.1: Common quantum error types.

9.2 The Bit-Flip Code (3-Qubit Repetition)

Encoding: 1 logical qubit into 3 physical qubits:

$$|0\rangle_L = |000\rangle\,, \qquad |1\rangle_L = |111\rangle \tag{9.1}$$

Circuit: CNOT(q_0, q_1), CNOT(q_0, q_2) encodes $\alpha|0\rangle + \beta|1\rangle \to \alpha|000\rangle + \beta|111\rangle$.
Syndrome extraction: Measure parity checks Z_1Z_2 and Z_2Z_3:

Error	**Syndrome** ($Z_1Z_2,\ Z_2Z_3$)	**Correction**
None	$(+1,+1)$	None
Qubit 1	$(-1,+1)$	X on qubit 1
Qubit 2	$(-1,-1)$	X on qubit 2
Qubit 3	$(+1,-1)$	X on qubit 3

Table 9.2: Bit-flip code syndrome table.

9.3 The Phase-Flip Code

Protects against Z errors using Hadamard basis encoding:

$$|0\rangle_L = |{+}{+}{+}\rangle\,, \qquad |1\rangle_L = |{-}{-}{-}\rangle \tag{9.2}$$

Corrects a single Z error using syndrome extraction in the X-basis.

9.4 The Shor Code (9-Qubit)

Concatenates bit-flip and phase-flip codes: 1 logical qubit in 9 physical qubits. Corrects **any** single-qubit error (X, Y, or Z). The first quantum error correcting code (Shor, 1995).

Exercises — Chapter 9

9.1. Build the bit-flip code in the Lab: $H(q_0)$ to create $|+\rangle$, then $\mathrm{CX}(q_0, q_1)$, $\mathrm{CX}(q_0, q_2)$. Add $X(q_1)$ as error. Apply correction. Does the result match?

9.2. Using the density matrix backend, apply depolarizing noise with $p = 0.1$ to $|+\rangle$. What is $\mathrm{Tr}(\rho^2)$ before and after?

Chapter 10

Cryptography: BB84 Quantum Key Distribution

10.1 The BB84 Protocol

BB84 (Bennett–Brassard, 1984) generates a shared secret key whose security is guaranteed by quantum mechanics.

Protocol:

1. **Alice** generates random bits and random bases (Z or X). She encodes: Z-basis uses $|0\rangle\,/\,|1\rangle$, X-basis uses $|+\rangle\,/\,|-\rangle$.
2. **Alice sends** qubits to Bob over a quantum channel.
3. **Bob** measures each qubit in a randomly chosen basis.
4. **Basis reconciliation:** Publicly compare bases (not bits). Keep bits where bases matched (∼50%).
5. **Eavesdropping check:** Sacrifice some bits to estimate error rate. If >11%, abort.
6. **Privacy amplification:** Hash remaining bits to get the final key.

10.2 Security Guarantee

An eavesdropper (Eve) who intercepts and re-sends qubits inevitably introduces errors: she doesn't know Alice's bases, and measuring in the wrong basis disturbs the state (no-cloning theorem prevents perfect copying). This disturbance is detectable in step 5.

Expected error rate under intercept-resend attack: 25% on matching-basis bits (Eve guesses wrong basis 50% of the time, causing 50% error when she does).

Exercises — Chapter 10

10.1. Simulate BB84 by hand for 8 qubits. Alice's bits: 10110100, bases: ZXZXZZXZ. Bob's bases: ZZZXXZXZ. Which bits survive?

10.2. If Eve intercepts all qubits with random bases, what is the expected error rate on surviving bits?

Chapter 11

Exercises and Problem Sets

These integrative exercises combine concepts from multiple chapters. Each set is designed for 30–60 minutes.

11.1 Problem Set A: Building Intuition (Beginner)

Problem Set A

A.1. Gate Explorer: Apply each single-qubit gate ($H, X, Y, Z, S, T, \sqrt{X}$) to $|0\rangle$. Record $P(0)$, $P(1)$, Bloch coordinates, and $|1\rangle$ amplitude. Which gates create superposition? Which only change phase?

A.2. Rotation Sweep: Apply $R_y(\theta)\,|0\rangle$ for $\theta = 0, \pi/6, \pi/4, \pi/3, \pi/2, 2\pi/3, \pi$. Plot $P(|1\rangle)$ vs θ. Derive that $P(|1\rangle) = \sin^2(\theta/2)$.

A.3. Interference Demo: Build H, $P(\theta)$, H for various θ. Show $P(|0\rangle) = \cos^2(\theta/2)$. What θ gives $P(|0\rangle) = 0$?

11.2 Problem Set B: Entanglement (Intermediate)

Problem Set B

B.1. Bell State Tomography: For $|\Phi^+\rangle$, measure in ZZ, XX, and YY bases. Record correlations $\langle ZZ\rangle$, $\langle XX\rangle$, $\langle YY\rangle$.

B.2. Entanglement Swapping: Create $|\Phi^+\rangle$ on (q_0, q_1) and (q_2, q_3). Bell-measure (q_1, q_2). What state are q_0, q_3 in? They never interacted!

B.3. Monogamy: Try to maximally entangle q_0 with both q_1 and q_2. Show it's impossible.

11.3 Problem Set C: Algorithms (Advanced)

Problem Set C

C.1. 3-Qubit Grover: Search for $|101\rangle$ in $N = 8$. Design oracle and diffusion. Compute optimal iteration count.

C.2. Phase Estimation: Estimate the eigenvalue of T ($e^{i\pi/4}$) using 3 counting qubits. What precision?

C.3. VQE Surface: Run VQE for H_2 at 6 bond lengths. Plot the potential energy curve.

11.4 Problem Set D: Build Your Own (Expert)

Problem Set D

D.1. Superdense Coding: Send 2 classical bits using 1 qubit. Verify all 4 messages.

D.2. Quantum Walk: 1D walk with coin qubit (H), 2-qubit position register. Run 4 steps. Compare to classical.

D.3. Error Threshold: Encode with 3-qubit bit-flip code. Vary depolarizing p. At what p does encoded perform worse than unencoded?

Appendix A

OpenQASM 2.0 Reference

A.1 Syntax

```
OPENQASM 2.0;               // Required version header
include "qelib1.inc";       // Standard gate library
qreg q[3];                  // 3-qubit quantum register
creg c[3];                  // 3-bit classical register
h q[0];                     // Hadamard on qubit 0
cx q[0], q[1];              // CNOT
rx(pi/4) q[2];              // Parameterized rotation
measure q -> c;             // Measure all qubits
```

A.2 Gate Mapping

QASM	tiny-qpu	Parameters	Qubits
`h`	H	None	1
`x / y / z`	$X/Y/Z$	None	1
`s / sdg`	$S/S^\dagger$	None	1
`t / tdg`	$T/T^\dagger$	None	1
`rx / ry / rz`	$R_x/R_y/R_z$	Angle (rad)	1
`cx`	CNOT	None	2
`cz`	CZ	None	2
`swap`	SWAP	None	2
`ccx`	Toffoli	None	3
`cswap`	Fredkin	None	3

Table A.1: QASM to tiny-qpu gate mapping.

A.3 Examples

Bell State:

```
OPENQASM 2.0;
include "qelib1.inc";
qreg q[2];
h q[0];
cx q[0], q[1];
```

3-Qubit QFT:

```
OPENQASM 2.0;
include "qelib1.inc";
qreg q[3];
h q[0];
crz(pi/2) q[1], q[0];
crz(pi/4) q[2], q[0];
h q[1];
crz(pi/2) q[2], q[1];
h q[2];
swap q[0], q[2];
```

Appendix B

Keyboard Shortcuts & Troubleshooting

B.1 Keyboard Shortcuts

Key	**Action**	**Context**
R	Run simulation	Any
S	Enter step mode	Any
C	Clear circuit	Any
Ctrl+Z	Undo last gate	Any
H	Select Hadamard	Not in text field
X / Y	Select Pauli-X / Y	Not in text field
M	Select Measure	Not in text field
→ / ←	Next / previous step	Step mode
Escape	Exit step mode	Step mode
Enter	Confirm parameter	Dialog
?	Open help	Any

Table B.1: Keyboard shortcuts.

B.2 Troubleshooting

Problem	**Cause**	**Solution**
App won't launch	Missing pywebview	`pip install pywebview`
ModuleNotFoundError	Missing Flask	`pip install flask`
No window appears	WebView2 issue	Use `--browser` flag
Simulation error	Invalid qubit index	Check gate targets fit qubit count
Bloch sphere empty	No simulation run	Click Run or press R
QASM import fails	Bad format	Must start with `OPENQASM 2.0;`
Port 8888 busy	Another process	Use `--port 9000`
Exe blocked	Windows Defender	Allow through firewall

Table B.2: Common issues and solutions.

tiny-qpu — Technical Manual & Quantum Computing Primer
Version 2.0 — February 2026
`github.com/SKBiswas1998/tiny-qpu`