Constructing quantum circuits

Turn a sequence of gates into a small, testable computational story.

16 min read Dr. Charith Jayasekara

By the end, you'll be able to

- Write a circuit contract with initial state, gates and readout basis.
- Compare a predicted distribution with a seeded finite-shot sample.

Start with a contract

Before adding gates, specify the initial state, qubit ordering, intended transformation and measured output. "Create correlated bits" is incomplete: it does not identify the quantum state or the measurement basis.

For a Bell-state circuit, use this contract:

- Input: $|00\rangle$, q0 leftmost.
- Gates: H on q0, then CX from q0 to q1.
- Exact output: $(|00\rangle + |11\rangle)/\sqrt{2}$.
- Z readout: 00 and 11 with probability $1/2$ each.

Verify at more than one level

First calculate the ideal state. Then compare the exact simulator distribution. Finally sample finite shots with a recorded seed. The first two check the model calculation; the third demonstrates sampling fluctuations.

The playground's Undo removes a gate from the circuit description and recomputes from the initial state. It is an editing operation, not a claim that a measured physical state can be reversed by deleting a line of history.

A deliberate defect

Move H to q1 but leave CX's direction unchanged. The control remains 0, so the result is $|0\rangle \otimes |+\rangle$, not a Bell state. Its nonzero Z outcomes are 00 and 01.

Repair the circuit and explain the root cause. This is more informative than simply trying gates until the bars look familiar.

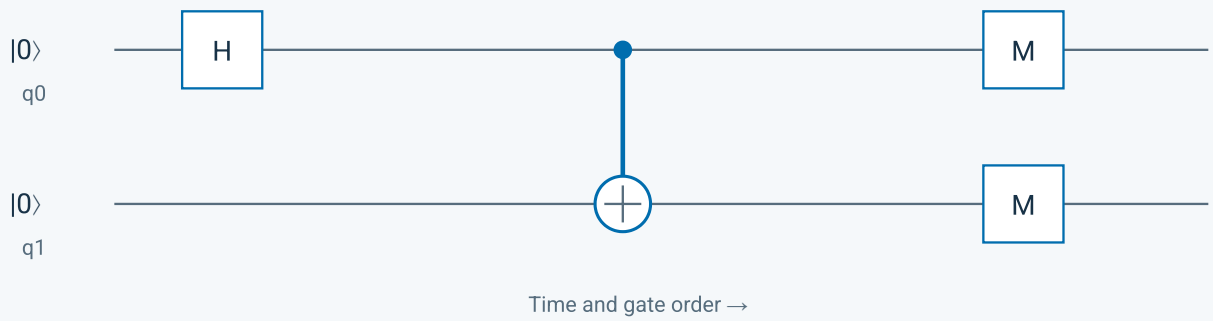
Extend

Ask whether your test would detect a relative-phase error. A Z gate after Bell preparation leaves the Z counts unchanged but changes the state. Add X-basis evidence using the entanglement lab.

A circuit diagram is a specification

WORKED DIAGRAM · F19

A circuit contract separates preparation, gates and readout



The input labels, gate order, control direction and measured wires are all part of the specification. The readout bits are ordered q0 then q1.

A complete circuit story identifies its inputs, transformations and readout. The diagram above specifies a Bell preparation, but an implementation also needs the bit-order convention and measurement basis. Without them, two correct-looking programs may report different strings or answer different experimental questions. Treat the circuit as a contract that another person can reproduce.

Separate the mathematical objective from the sample you happen to observe. The objective might be an exact state or a distribution. A finite sequence of counts is evidence about that objective under a model, not the objective itself. This distinction prevents a test suite from accepting a wrong circuit merely because a small random sample looked plausible.

Worked trace: build a Bell contract

Start with the column one, zero, zero, zero. H on q0 produces equal amplitudes at 00 and 10. CX from q0 to q1 moves the 10 entry to 11. The final vector has equal positive entries at 00 and 11 and zeros elsewhere. Its norm is one, and its Z probabilities are one half on each agreeing pair.

Now append H to both qubits for an X-basis readout. Expanding the two terms shows that the cross outcomes cancel and the agreeing outcomes reinforce. The ideal distribution is again half 00 and half 11. The amplitude table after those readout gates describes a different stage from the original Bell preparation, even though the numerical vector is the same for this particular Bell state.

If you instead append H only to q0, all four Z outcomes become equally probable, with one relative sign. A test that expects only agreeing outcomes must therefore specify both readout rotations. Small changes to a circuit's final gates can change the interpretation of the entire result.

Plan three levels of checks

First check local algebra. Verify known gate columns, norms and inverse identities independently of random sampling. Second check the composed ideal state or distribution against a hand-worked example. Third collect finite-shot counts and compare their uncertainty with the predicted distribution. A sampling test cannot replace a phase-sensitive state test, and an exact state test does not validate a physical device's calibration.

Include cases that could expose your likely mistake. For CX, use an asymmetric basis input to catch reversed wires, and a superposed control to catch lost coherence. For H, use both zero and one so a missing minus sign is visible. For phase gates, add a final H to make their effect observable in probabilities. Good tests distinguish plausible wrong implementations from the intended one.

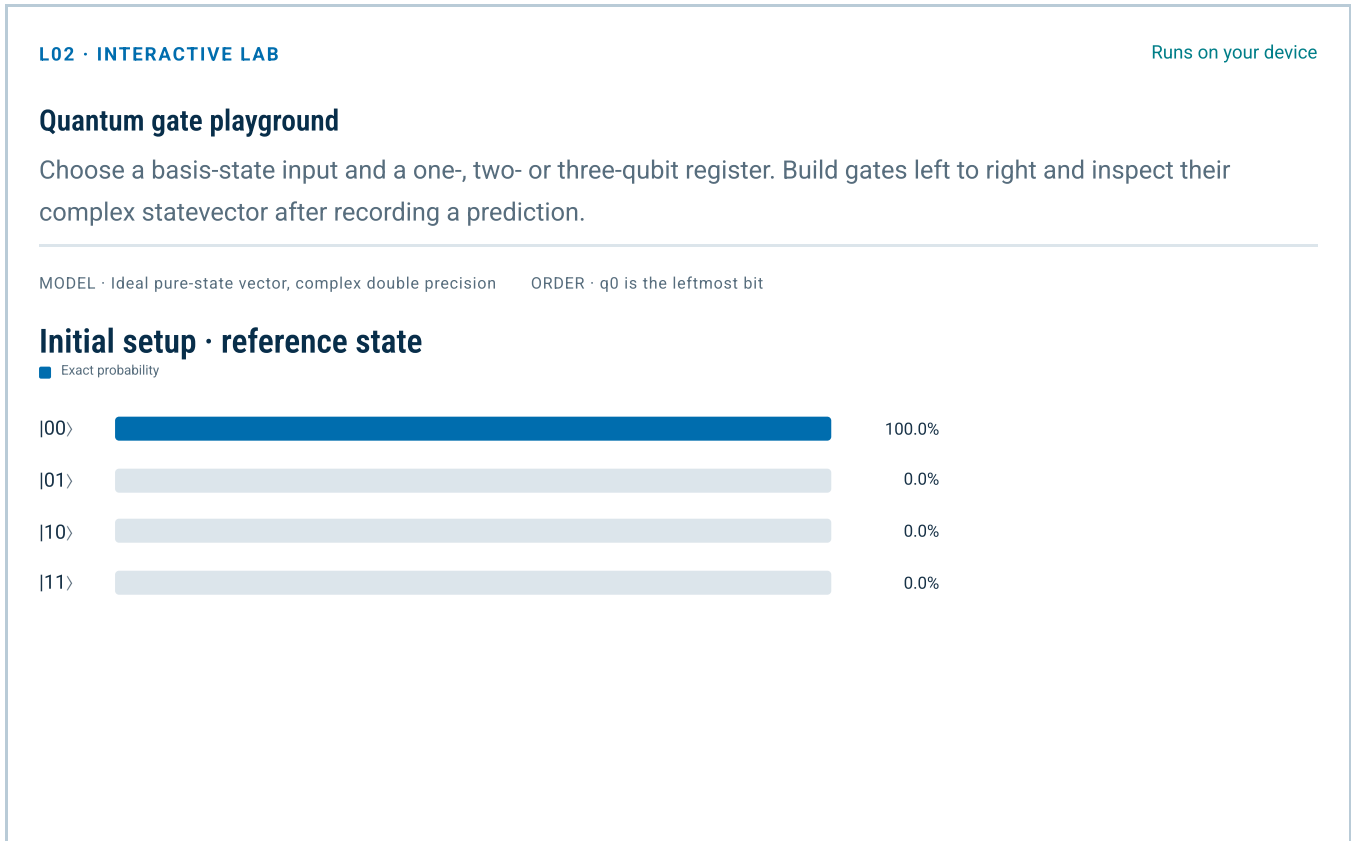
Keep an experiment record

Record the input, gate order, wire indices, measurement basis, model assumptions, shot budget and seed. If you compare tools, also record their bitstring and Fourier conventions. A seed provides reproducibility for this sampler implementation; it does not imply that another random-number algorithm must return identical counts.

When the result disagrees, localise the first divergence. Check input preparation, then the state after each gate, then probability conversion and sampling. Looking only at the last histogram makes sign, ordering and statistical errors look similar. The full amplitude table gives you a diagnostic view that is available in simulation but generally unavailable as direct hardware telemetry.

Build the contract in the playground

The default input is 00. Add H to q0, then CX from q0 to q1. Write the four expected amplitudes in order, reveal and compare. Run measurements only after checking the exact model. An ideal sample may contain unequal counts of 00 and 11, but it should not contain 01 or 10 because their probabilities are zero.



Statevector before readout · ideal simulation

Basis	Amplitude a + bi	Magnitude	Phase (rad)	Probability
|00⟩	`1 + 0i`	1.000000	0.000000	100.0%
|01⟩	`0 + 0i`	0.000000	undefined	0.0%
|10⟩	`0 + 0i`	0.000000	undefined	0.0%
|11⟩	`0 + 0i`	0.000000	undefined	0.0%

Signs and complex phases are part of the state. Phase is undefined for a zero amplitude. A shared global phase changes no probabilities; relative phases can affect a later operation. Values are rounded to six decimal places.

Prediction to try: For the chosen input and circuit, predict the nonzero amplitudes, their signs and the measured bitstrings.

Challenge: Compare H|0⟩ with H|1⟩ using one qubit. Their Z probabilities agree; their amplitude signs do not. Add a second H to distinguish them.

Open the live experiment studio

Now undo CX, leaving the first H. Predict the new distribution before revealing it. Restore CX and add H to q0 alone. Predict all four amplitudes and explain the sign pattern. This controlled sequence makes the relationship between circuit editing and state evolution explicit. Reset returns the input and clears all gates rather than preserving your previous partial experiment.

Practice: diagnose a misleading success

A circuit intended to prepare a Bell state returns only agreeing Z outcomes. A student declares it verified. Give an alternative preparation that passes this test, then propose an additional check that distinguishes it from the ideal Bell state.

Check your reasoning

A classical equal mixture of 00 and 11 has the same Z distribution. Measure both wires in X on fresh preparations: the Bell state still has agreeing results, while that mixture gives all four pairs equally. Alternatively, an ideal inverse Bell circuit returns the Bell state to 00 but does not do so deterministically for the mixture. A practical verification would add statistical uncertainty and device-error analysis.

Reference: IBM Quantum — Bits, gates, and circuits.

Check your understanding

PAUSE & THINK

Which record makes a sampled circuit result reproducible here?

- [ ] Only the seed
- [ ] Input, ordered gates, wire convention, basis, model, shots and seed
- [ ] Only the final histogram image

Answer: Input, ordered gates, wire convention, basis, model, shots and seed. The seed fixes sampling only after the distribution and implementation conventions are specified.