Introduction to quantum gates

Small operations on amplitudes become the building blocks of a circuit.

16 min read Dr. Charith Jayasekara

By the end, you'll be able to

- Trace the order of operations in a circuit.
- Distinguish a gate's effect on amplitudes from its effect on measurement probabilities.

An operation, not an assignment

An ideal quantum gate applies a unitary matrix to a state. X exchanges amplitudes. Z changes the sign of the 1 component. H mixes components so they can reinforce or cancel.

If H is applied first and Z second, the resulting state is $ZH|\psi\rangle$. The rightmost matrix acts first. Circuit diagrams here read left to right, so follow the physical order when translating to algebra.

A small gate dictionary

Gate	Action on a one-qubit state $(\alpha, \beta)^T$
X	$(\beta, \alpha)^T$
Z	$(\alpha, -\beta)^T$
H	$(\alpha + \beta, \alpha - \beta)^T/\sqrt{2}$
S	$(\alpha, i\beta)^T$
T	$(\alpha, e^{i\pi/4}\beta)^T$

Predict, then build

Apply H to q0 and CX from q0 to q1. Predict which two basis outcomes survive. Use Undo to check the circuit one step at a time.

Order can matter

On $|0\rangle$, H followed by Z gives $|-\rangle$. Z followed by H gives $|+\rangle$. Their Z probabilities match, so add a final H to reveal the difference.

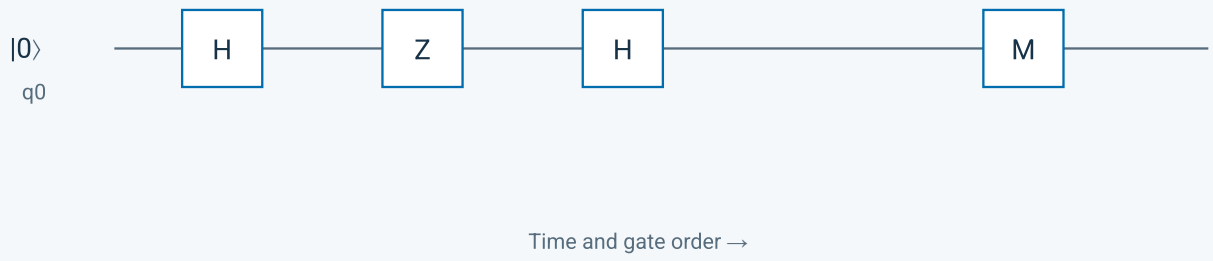
Do not infer that two circuits are equivalent merely because they give the same counts for one input in one basis. A gate is a transformation with a contract over its allowed inputs.

Challenge: find two gates in the table that square to I. Then explain why S is not one of them.

A gate acts on amplitudes, not on printed probabilities

WORKED DIAGRAM · F15

Trace a circuit from left to right



On a zero input, the first H prepares plus, Z changes it to minus, and the second H gives one. Algebra writes the same operation HZH, with the rightmost factor acting first.

The diagram's H–Z–H circuit is a complete transformation of one wire. H first mixes the zero input into two components. Z reverses one sign. The final H combines the components again and produces one. If a gate implementation updates only probabilities, it cannot reproduce this calculation because it has discarded the relative sign before the last step.

An ideal gate is independent of whether you happen to know its input. Its matrix rule applies linearly to every state. In the simulator, the complex statevector is updated directly. On hardware, control pulses implement an approximation to the desired physical transformation. The mathematical gate is a specification, while a calibrated implementation has errors and timing costs that must be characterised separately.

Worked example: H, S, H

Begin with zero. H creates equal positive amplitudes. S multiplies the one amplitude by i, leaving the zero component unchanged. The last H gives amplitudes one half plus i over two and one half minus i over two. Their probabilities are equal, but the output is not plus: its two complex entries have different phases.

Now replace S with Z. The final amplitudes become zero and one. Replace S with the identity and the final amplitudes become one and zero. The gate between the H operations controls interference. These three trials show why a phase gate can be useful even though it never changes immediate Z probabilities on its own.

To check a circuit, write a small trace table with one row after each gate and one column per basis label. Preserve real and imaginary parts until the end. Calculate probabilities only at the readout stage, unless you are explicitly comparing intermediate model states. A trace table localises an error: if the first mismatch occurs after S, you need not blame the final H or the sampler.

Gates on different wires

In a multi-qubit statevector, a one-qubit gate acts on pairs of entries whose basis labels differ only at its target wire. H on q0 mixes 00 with 10 and 01 with 11. H on q1 instead mixes 00 with 01 and 10 with 11. This pairing follows from the tensor-product definition and the leftmost-q0 convention.

Gates on separate wires commute when they act independently. H on q0 followed by X on q1 gives the same joint transformation as the reversed order. Gates on the same wire need not commute. Controlled gates can also couple the roles of different wires, so moving an operation through a CX requires an algebraic justification rather than a visual shortcut.

The playground recomputes your full displayed circuit from the chosen input whenever you edit it. Changing the basis-state selector does not add a preparation gate at the end; it changes the input contract. Choosing a target does not itself transform the state; pressing a gate button appends an operation on that target. These distinctions help you build reproducible comparisons.

A useful gate set is not a complete cost model

A small collection of gates can express many circuits. Exact representation and approximate synthesis are different questions, especially for continuously variable rotations. A hardware backend may support a different native set and require compilation. Two short abstract circuits can therefore have different depths or error exposure after mapping to a device.

At this stage, focus on functional correctness: what state should the circuit produce under the ideal convention? Later compare native gate counts, connectivity, precision and fault-tolerant costs. Reporting only an abstract number of gates can hide expensive synthesis or routing. Likewise, measuring how quickly the browser updates a tiny statevector says little about quantum hardware execution time.

Build and verify one circuit

Use two qubits with input 00 and target q0. Add H, Z and H to that wire, in order. Predict the four-entry output and reveal. It should be concentrated on 10, because q0 is the leftmost bit and q1 stays zero. Reset and repeat with target q1; the output should instead be 01.

L02 · INTERACTIVE LAB

Runs on your device

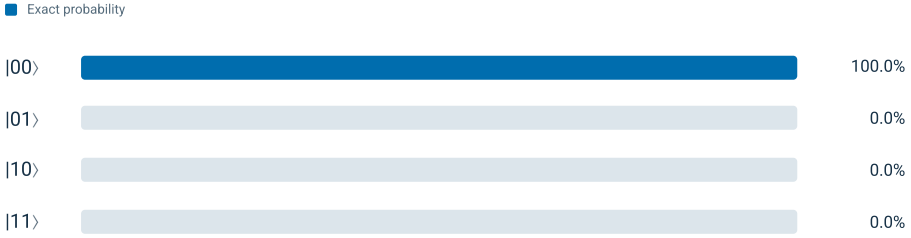
Quantum gate playground

Choose a basis-state input and a one-, two- or three-qubit register. Build gates left to right and inspect their complex statevector after recording a prediction.

MODEL · Ideal pure-state vector, complex double precision

ORDER · q0 is the leftmost bit

Initial setup · reference state



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

For an additional check, prepare plus on q0 and compare X with Z. X leaves plus unchanged, while Z turns it into minus. Both have equal Z probabilities, so inspect the signed amplitudes and add H before readout when you want the difference to appear in the histogram. Record the input and gate order with your explanation.

Practice: find a wrong translation

A diagram applies H first and Z second, but a student writes the output as HZ acting on zero. Evaluate both orders and identify the correct algebraic expression.

Check your reasoning

The correct expression is ZH acting on zero, because the rightmost matrix acts first. It produces minus. HZ acting on zero produces plus, since Z leaves zero unchanged. Circuit time runs left to right in these diagrams, while matrix products act on a column from right to left. Translating the entire sequence before multiplying avoids reversing the computation accidentally.

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

Check your understanding

PAUSE & THINK

H is applied first, then Z. Which expression represents the output?

- ☐ HZ|ψ⟩
- ☐ ZH|ψ⟩
- ☐ H+Z|ψ⟩

Answer: ZH|ψ⟩. The rightmost matrix acts first on a column. Circuit time here runs left to right.