Deutsch–Jozsa Algorithm

Extend phase kickback to distinguish constant and balanced functions, while keeping the promise and classical comparison explicit.

Core 21 min lesson Dr. Charith Jayasekara

By the end, you'll be able to

- State the constant-or-balanced promise precisely.
- Calculate the all-zero output amplitude as an average of signs.
- Compare exact deterministic and bounded-error randomised classical access.

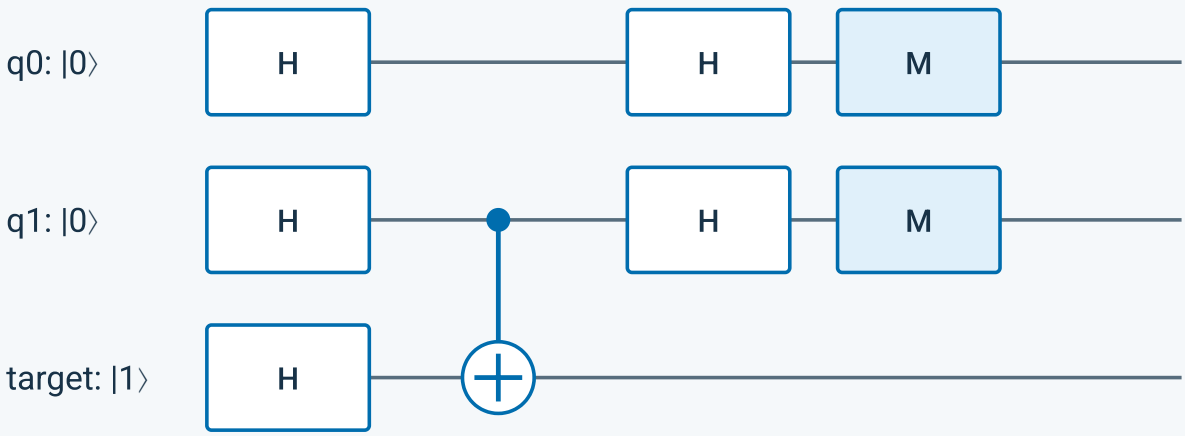
Ask a global question under a promise

Deutsch–Jozsa extends the agreement question to functions of many input bits. The function is promised to be either constant or balanced. Constant means every input has the same output. Balanced means exactly half the inputs give zero and half give one. The goal is to identify which case holds, without learning the entire truth table.

Read Deutsch's algorithm first, and review tensor products. The circuit is the same phase-kickback idea applied to an n-qubit input register. The proof focuses on one final amplitude: that of the all-zero bitstring.

WORKED DIAGRAM · A03

Deutsch–Jozsa: a balanced last-bit function



The target starts in |1⟩ and becomes |−⟩. The CNOT gates implement this example's reversible Boolean oracle. Final input readout is 01.

The setup and the comparison

Access is through a reversible function oracle that XORs the Boolean output into a target. The promise excludes functions with, for example, three zeros and one one on a two-bit domain. The output is one classical classification label. In the ideal quantum model, one oracle call suffices for certainty under the promise.

An exact deterministic classical algorithm can be forced to inspect more than half the truth table before distinguishing the two cases: after observing the same value on half the inputs, the unseen half could still make the function balanced. The worst-case exact deterministic count is $2^{n-1}+1$. Randomised classical sampling changes the comparison sharply: a small number of random queries detects a balanced function with high probability. The exponential deterministic contrast should not be described as an exponential advantage over every classical method.

Checkpoint 2

PAUSE & THINK

Which classical comparison gives the familiar exponential exact-query contrast?

- ☐ Exact deterministic algorithms
- ☐ Every randomised algorithm at fixed error
- ☐ Any classical program regardless of its access model

Answer: Exact deterministic algorithms. The deterministic exact contrast is exponential; allowing classical bounded error changes that comparison substantially.

The procedure

Prepare n input qubits in zero and one target qubit in one. H on every wire creates a uniform input and a minus target. Query once. Phase kickback multiplies input x by the sign determined by f(x). Apply H to every input qubit, measure them, and inspect whether all recorded bits are zero. All-zero means constant; any nonzero string means balanced, under the promise.

Use a bundle for the n input wires and a separate target wire. Show the target factored out as minus after the query. The output box should say “test all-zero,” not “read the truth table.” This makes the limited classical output explicit.

After phase kickback, every input label still has the same probability, but its amplitude has acquired the function’s sign.

$$|\psi_f\rangle = \frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} (-1)^{f(x)}|x\rangle.$$

Here n is the number of input bits. The unchanged minus target is factored out.

Worked two-bit example

Take inputs in order 00, 01, 10 and 11, and let f(x) be the last bit. Its values are zero, one, zero, one, so it is balanced. Before the oracle the four amplitudes are all one half. After phase kickback they are one half, negative one half, one half

and negative one half.

Applying H to both input qubits is a four-dimensional Hadamard transform. The resulting amplitude is one on 01 and zero elsewhere. The output is therefore nonzero with certainty. This particular linear function yields a particular deterministic string. The general theorem does not require every balanced function to have that same distribution over nonzero strings.

Now take constant one. The four post-oracle amplitudes are all negative one half. After the final H layer, only 00 has a nonzero amplitude, negative one. The probabilities identify constant even though the global sign differs from constant zero. This is the same physical equivalence encountered in the one-bit algorithm.

For the last-bit function, the signs match the Hadamard pattern for label 01.

$$f(x_0x_1) = x_1, \qquad \frac{1}{2}\begin{pmatrix} 1 \\ -1 \\ 1 \\ -1 \end{pmatrix} \xrightarrow{H^{\otimes 2}} |01\rangle.$$

The column uses numeric label order $00, 01, 10, 11$, with x_0 on the left. A nonzero result establishes the balanced case under the promise.

Why the all-zero amplitude is enough

The amplitude for any output z is a signed sum over inputs. For z equal to all-zero, the extra Hadamard signs are all positive, so the expression simplifies to the average of the oracle signs:

$$\alpha_{0^n} = \frac{1}{2^n}\sum_x (-1)^{f(x)}.$$

For a constant function, every term agrees, giving amplitude positive or negative one. For a balanced function, half the terms are positive and half negative, giving zero. Squaring the magnitude gives the advertised probabilities. We never needed to determine every other amplitude to prove the classification rule.

This is a useful proof strategy. Find a statistic whose value is forced by the promise, then design interference to place that statistic in a measurable amplitude. The algorithm does not calculate a conventional average and print it as a real number; the average appears as one amplitude of a quantum state, and its special promised values make the final test exact.

Each final amplitude is an average of signs. At all-zero, there is no extra Hadamard sign to include.

$$\alpha_y = \frac{1}{2^n}\sum_x (-1)^{f(x)+x\cdot y}, \qquad x\cdot y = \bigoplus_{j=0}^{n-1} x_j y_j.$$

For $y = 0^n$ this becomes the chapter's all-zero formula. Constant functions give magnitude one there; balanced functions give zero. The binary dot product is evaluated modulo two in the exponent.

Checkpoint 1

PAUSE & THINK

A two-bit function has three zeros and one one. What is its all-zero output probability?

- ☐ 0
- ☐ 1/4
- ☐ 1

Answer: 1/4. The amplitude is 1/2 and the probability is 1/4. This function violates the promise, so the usual classification guarantee does not apply.

What if the promise fails?

Suppose a two-bit function gives one on only one of its four inputs. The oracle-sign sum is two, so the all-zero amplitude is one half and its probability is one quarter. The circuit can return zero or a nonzero string. Neither outcome supplies the original promised guarantee because the actual function belongs to neither class.

An algorithm can be physically well-defined on a non-promised input while its advertised interpretation fails. A promise is part of the problem statement, not an optional assumption added after observing a convenient answer. This distinction reappears in Simon's algorithm and many hidden-structure problems.

Cost and practical interpretation

Outside the oracle, the circuit uses a number of H gates linear in n, plus preparation and readout. A full implementation must also build the reversible oracle, manage workspace and handle errors. A short query circuit can contain a computationally expensive oracle. A toy simulation validates the state evolution for small instances; its browser timing does not benchmark a quantum processor.

The pedagogical value is the explicit cancellation of a global pattern. Bernstein–Vazirani reuses the same transform to recover a hidden linear string, and Simon uses Fourier sampling to obtain constraints on a hidden period. Their different promises and outputs explain why superficially similar circuits solve different problems.

Predict and test

Use two or three input qubits and the promised oracle selector. Predict the all-zero probability before using Reveal in Predict & test mode. The built-in balanced function is the last input bit; other balanced functions can have different nonzero output patterns.

L05 · INTERACTIVE LAB

Runs on your device

Deutsch–Jozsa explorer

A constant-or-balanced Boolean oracle acts by phase kickback, followed by H on each input qubit. The unchanged |−〉 ancilla is factored out. One input qubit gives Deutsch's original case.

MODEL · Ideal pure-state vector, complex double precision ORDER · q0 is the leftmost bit

Initial setup · reference state

Exact probability



Statevector before readout · ideal simulation

Basis	Amplitude a + bi	Magnitude	Phase (rad)	Probability
|00〉	`0 + 0i`	0.000000	undefined	0.0%
|01〉	`1 + 0i`	1.000000	0.000000	100.0%
|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: Predict whether the all-zero input-register outcome can occur for the selected promised oracle.

Challenge: Constant-zero and constant-one differ by a global sign before the final H gates. Explain why their measured probabilities agree.

Open the live experiment studio

Chapter summary

- The constant-or-balanced promise supplies the meaning of the final bitstring.
- Balanced signs cancel exactly at the all-zero output.
- The exact deterministic classical comparison differs from a randomised bounded-error comparison.

A common misconception

Any nonconstant function is a promised balanced function. Balanced means exactly half zero and half one. Other nonconstant functions fall outside the two-case promise.

Exercises

1. A three-bit promised function is constant one. What is the final input-register state, including its global sign?
2. For a two-bit non-promised function with exactly one output equal to one, calculate the all-zero probability. Why is a zero observation not evidence that it is constant?

Worked answers

1. Every input amplitude receives a negative sign. The final H layer gives negative 000. The probability of 000 is one, and the common minus is physically irrelevant.
2. Three positive signs and one negative sign sum to two. Dividing by four gives amplitude one half, hence probability one quarter. The function violates the promise, so the classification rule does not apply; zero is possible without constancy.

Check your understanding

PAUSE & THINK

A two-bit function has three zero outputs and one one output. What is its all-zero circuit probability?

- [ ] 0
- [ ] 1/4
- [ ] 1

Answer: 1/4. The sign average is (3−1)/4=1/2, whose square is 1/4. This function violates the promise, so the usual classification guarantee does not apply.

Sources

IBM's Deutsch–Jozsa lesson develops the promised task and its Fourier-sampling relatives.