# Quantum Computing

## Unit 4: Introductory Quantum Algorithms

Dr. Gopal Chandra Jana

Dept. of CSE, Sharda University

Quantum Computing (CSE063) — Unit 4

August 16, 2026

Course Page:

https://www.gcjana.in/courses/shardauniversity/2501/quantum-computing/

# Outline

# Learning Outcomes of Unit 4

**After this unit, a student will be able to...**

1. **Contrast** probabilistic and quantum algorithms and explain the role of *interference*.
2. **Describe** the oracle (query) model and the standard oracle $U_f$.
3. **Apply** the *phase kick-back* trick to turn a bit-oracle into a phase-oracle.
4. **Derive** the Deutsch and Deutsch–Jozsa algorithms and their single-query advantage.
5. **Explain** Simon's algorithm and its exponential quantum speed-up.
6. **Compare** classical and quantum query complexity honestly (deterministic vs. randomised).

**Mapping to the Syllabus**

**A.** Probabilistic vs. quantum algorithms; phase kick-back. **B.** The Deutsch and Deutsch–Jozsa algorithms. **C.** Simon's algorithm.

# Probabilistic vs. Quantum Algorithms

| Aspect | Probabilistic | Quantum |
|---|---|---|
| State | probability vector ($p_i \geq 0$, $\sum p_i = 1$) | amplitude vector ($c_i \in \mathbb{C}$, $\sum \lvert c_i \rvert^2 = 1$) |
| Evolution | stochastic matrices | unitary matrices |
| Entries | non-negative | complex (can be *negative*) |
| Key phenomenon | randomness | **interference** (amplitudes cancel) |
| Read-out | sample a string | Born-rule measurement $\Pr = \lvert c_i \rvert^2$ |

**The one idea that matters**

Probabilities only *add*; amplitudes can **cancel**. Quantum algorithms are engineered so that the amplitudes of *wrong* answers destructively interfere.

# The Oracle (Query) Model

**Standard oracle**

A function $f : \{0,1\}^n \to \{0,1\}^m$ is given as a *black box*. To be reversible it is implemented as the unitary

$$U_f |x\rangle |y\rangle = |x\rangle |y \oplus f(x)\rangle .$$

- $U_f$ is a permutation (hence unitary) and its own inverse: $U_f^2 = I$.
- We measure cost by the number of **queries** to $U_f$ (query complexity).
- Setting $|y\rangle = |0\rangle$ gives $|x\rangle |f(x)\rangle$.

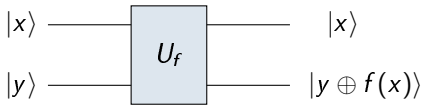


Figure: the reversible oracle $U_f$.

# Interference — The Quantum Advantage

## A one-line demonstration

Apply $H$ twice to $|0\rangle$:

$$|0\rangle \xrightarrow{H} \frac{|0\rangle+|1\rangle}{\sqrt{2}} \xrightarrow{H} |0\rangle\,.$$

The two paths to $|1\rangle$ carry amplitudes $+\frac{1}{2}$ and $-\frac{1}{2}$ and **cancel**.

- A classical random walk could never make a possibility vanish.
- Every quantum algorithm here is a careful arrangement of such cancellations.

## Not "parallel universes"

A superposition over $2^n$ inputs is *not* free parallelism: measurement returns one string. The power comes from making useful amplitudes **reinforce** before we measure.

# Phase Kick-Back — The Key Trick

## Put the target in $|-\rangle$

With $|-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$,

$$U_f\,|x\rangle\,|-\rangle = |x\rangle\,\tfrac{1}{\sqrt{2}}\big(\,|f(x)\rangle - |1 \oplus f(x)\rangle\,\big) = \boxed{(-1)^{f(x)}\,|x\rangle\,|-\rangle}.$$

## Case check

- $f(x) = 0$: $\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) = +\,|-\rangle$.
- $f(x) = 1$: $\frac{1}{\sqrt{2}}(|1\rangle - |0\rangle) = -\,|-\rangle$.

## What happened

The oracle's *output* was "kicked back" as a **phase** $(-1)^{f(x)}$ on the *input* register; the target $|-\rangle$ is untouched. A bit-oracle became a **phase-oracle** $|x\rangle \mapsto (-1)^{f(x)}\,|x\rangle$.

# Phase Kick-Back — The General Picture

**Eigenstates absorb, phases escape**

If $U|\psi\rangle = e^{i\varphi}|\psi\rangle$ (eigenstate), then a *controlled-U* leaves the target $|\psi\rangle$ unchanged and imprints $e^{i\varphi}$ on the control:

$$|1\rangle|\psi\rangle \mapsto e^{i\varphi}|1\rangle|\psi\rangle\,.$$

**Why it is so useful**

Phases sit in amplitudes, where they can **interfere**. Kick-back moves the answer to exactly where interference can act on it.

- Here $|-\rangle$ is the $(-1)$-eigenstate of the bit-flip, giving $(-1)^{f(x)}$.
- The same mechanism powers *phase estimation* and Shor's algorithm (later units).

# Quick Check — Round 1 (Model & Kick-Back)

**[Q1]**

Give the action of the standard oracle $U_f$ on $|x\rangle\,|y\rangle$.

**[Q2]**

Show that $U_f\,|x\rangle\,|-\rangle = (-1)^{f(x)}\,|x\rangle\,|-\rangle$.

**[Q3]**

In one sentence, what makes quantum algorithms more powerful than probabilistic ones?

**[Q4] True/False**

"$U_f$ is unitary and equal to its own inverse."

Think for a minute — answers on the next slide.

# Answers — Round 1

**[A1]**

$U_f |x\rangle |y\rangle = |x\rangle |y \oplus f(x)\rangle$.

**[A2]**

$U_f |x\rangle \frac{|0\rangle - |1\rangle}{\sqrt{2}} = |x\rangle \frac{|f(x)\rangle - |1 \oplus f(x)\rangle}{\sqrt{2}}$, which equals $+|-\rangle$ if $f(x) = 0$ and $-|-\rangle$ if $f(x) = 1$, i.e. $(-1)^{f(x)} |x\rangle |-\rangle$.

**[A3]**

Amplitudes can be negative/complex and **interfere** (cancel), which classical probabilities cannot.

**[A4]**

**True.** $U_f$ is a permutation of basis states (unitary) and applying it twice restores $y$: $U_f^2 = I$.

# Deutsch's Problem

## The task

Given $f : \{0,1\} \to \{0,1\}$ as an oracle, decide whether $f$ is

- **constant** ($f(0) = f(1)$), or
- **balanced** ($f(0) \neq f(1)$),

i.e. compute the single bit $f(0) \oplus f(1)$.

## Quantum promise

A single quantum query to $U_f$ suffices — thanks to phase kick-back and interference.

## Classical cost

You must learn *both* $f(0)$ and $f(1)$: **2 queries**.

# The Deutsch Algorithm

**Steps**

1. Start $|0\rangle\,|1\rangle$.
2. $H \otimes H$: $\quad |+\rangle\,|-\rangle$.
3. $U_f$ (kick-back): $\frac{1}{\sqrt{2}} \sum_x (-1)^{f(x)}\,|x\rangle\,|-\rangle$.
4. $H$ on the top qubit.
5. Measure the top qubit.

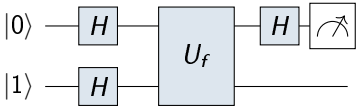


Figure: one oracle query decides constant vs. balanced.

# Deutsch — Why One Query Suffices

## Track the top register (bottom stays $|-\rangle$)

After the oracle, the top qubit is (up to the global phase $(-1)^{f(0)}$)

$$\tfrac{1}{\sqrt{2}}\big(\,|0\rangle + (-1)^{f(0)\oplus f(1)}\,|1\rangle\,\big).$$

Apply $H$ and let $a = f(0) \oplus f(1)$:

$$\tfrac{1}{2}\big[(1+(-1)^a)\,|0\rangle + (1-(-1)^a)\,|1\rangle\,\big] = \begin{cases} |0\rangle\,, & a = 0 \text{ (constant)} \\ |1\rangle\,, & a = 1 \text{ (balanced).} \end{cases}$$

Measuring the top qubit returns $f(0) \oplus f(1)$ with certainty — in a single query.

# The Deutsch–Jozsa Problem

**The task (with a promise)**

Given $f : \{0,1\}^n \to \{0,1\}$ promised to be *either*

- **constant** (same value on all $2^n$ inputs), *or*
- **balanced** (0 on exactly half the inputs, 1 on the other half),

decide which.

**Classical (deterministic) cost**

In the worst case you may need $2^{n-1} + 1$ queries to be *certain*.

**Quantum cost**

**One** query, with certainty.

# The Deutsch–Jozsa Algorithm

**Steps**

1. Start $|0\rangle^{\otimes n}|1\rangle$.
2. $H^{\otimes n} \otimes H$.
3. $U_f$ (kick-back).
4. $H^{\otimes n}$ on the top register.
5. Measure the top $n$ qubits.

*All zeros $\Rightarrow$ constant; anything else $\Rightarrow$ balanced.*

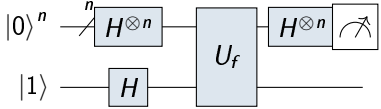


Figure: same shape as Deutsch, scaled to $n$ qubits.

# Deutsch–Jozsa — The Analysis

**Use $H^{\otimes n}|x\rangle = \frac{1}{\sqrt{2^n}} \sum_z (-1)^{x\cdot z} |z\rangle, \quad x \cdot z = \bigoplus_i x_i z_i$**

Before the final Hadamards the top register is $\frac{1}{\sqrt{2^n}} \sum_x (-1)^{f(x)} |x\rangle$. Afterwards the amplitude of $|0\cdots 0\rangle$ is

$$\frac{1}{2^n} \sum_{x \in \{0,1\}^n} (-1)^{f(x)}.$$

**Constant $f$**

Every term is $+1$ or every term $-1$: amplitude $= \pm 1 \Rightarrow$ outcome $|0\cdots 0\rangle$ with probability 1.

**Balanced $f$**

Half the terms are $+1$, half $-1$: they **cancel**, amplitude $= 0 \Rightarrow |0\cdots 0\rangle$ is *never* seen.

# Query Complexity — An Honest Comparison

| Problem | Classical queries | Quantum queries |
|---|---|---|
| Deutsch | 2 (deterministic) | 1 |
| Deutsch–Jozsa | $2^{n-1}+1$ (deterministic, exact) | 1 |
| Simon (next) | $\Theta(2^{n/2})$ (even with randomness) | $O(n)$ |

**The important caveat**

Deutsch–Jozsa's exponential gap is against *exact, deterministic* classical algorithms. A **randomised** classical algorithm decides D–J with only $O(1)$ queries and tiny error probability.

**Why Simon is special**

Simon's speed-up survives *even against randomised* classical algorithms — the first such natural exponential separation, and the direct inspiration for Shor.

# Quick Check — Round 2 (Deutsch & Deutsch–Jozsa)

**[Q1]**

What single bit does the Deutsch algorithm compute, and in how many queries?

**[Q2]**

In Deutsch–Jozsa, what measurement outcome indicates a *constant* function?

**[Q3]**

Write the amplitude of $|0\cdots0\rangle$ after the final Hadamards in Deutsch–Jozsa.

**[Q4]**

Is Deutsch–Jozsa an exponential speed-up over *randomised* classical algorithms? Explain briefly.

# Answers — Round 2

**[A1]**

It computes $f(0) \oplus f(1)$ (constant vs. balanced) in **one** query.

**[A2]**

The all-zeros string $|0 \cdots 0\rangle$; any other outcome means the function is *balanced*.

**[A3]**

$\frac{1}{2^n} \sum_x (-1)^{f(x)}$ — equal to $\pm 1$ (constant) or 0 (balanced).

**[A4]**

**No.** Against *deterministic* algorithms yes, but a randomised classical algorithm solves it in $O(1)$ queries with high probability. The true randomised-hard example is *Simon's*.

# Simon's Problem

## The task (hidden-string promise)

Given $f : \{0,1\}^n \to \{0,1\}^n$ promised to be **two-to-one** with a hidden $s \neq 0^n$ such that

$$f(x) = f(y) \iff y = x \text{ or } y = x \oplus s.$$

**Find** the period $s$.

## Classical cost

By a birthday-style argument, $\Theta(2^{n/2})$ queries are needed even *with randomness* — exponential in $n$.

## Quantum cost

$O(n)$ queries (expected), plus polynomial post-processing — an **exponential** speed-up.

# Simon's Algorithm — The Circuit

**Steps (one round)**

1. Start $|0\rangle^n |0\rangle^n$.
2. $H^{\otimes n}$ on the input register: $\frac{1}{\sqrt{2^n}} \sum_x |x\rangle |0\rangle$.
3. $U_f$: $\frac{1}{\sqrt{2^n}} \sum_x |x\rangle |f(x)\rangle$.
4. (Implicitly) measure the output register.
5. $H^{\otimes n}$ on the input register.
6. Measure the input register $\rightarrow$ a string $z$.

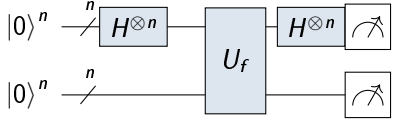


Figure: prepare, query, interfere, measure — then repeat.

# Simon's Algorithm — The Analysis

## Collapse, then interfere

Observing $f(x_0)$ in the output collapses the input to the two pre-images:

$$\tfrac{1}{\sqrt{2}}\big(\,|x_0\rangle + |x_0 \oplus s\rangle\,\big).$$

Applying $H^{\otimes n}$ gives $\dfrac{1}{\sqrt{2^{n+1}}}\displaystyle\sum_z (-1)^{x_0 \cdot z}\big[1 + (-1)^{s \cdot z}\big]\,|z\rangle$.

## The interference

$$1 + (-1)^{s \cdot z} = \begin{cases} 2, & s \cdot z = 0 \\ 0, & s \cdot z = 1. \end{cases}$$

## Outcome

We measure a *uniformly random* $z$ satisfying $\boxed{s \cdot z = 0 \pmod 2}$ — one linear equation about the unknown $s$.

# From Samples to $s$ — Linear Algebra over $\mathrm{GF}(2)$

**Collect equations**

Each run gives a random $z$ with $s \cdot z = 0$. Repeat to gather $n-1$ *linearly independent* equations

$$z^{(1)} \cdot s = 0, \ \ldots, \ z^{(n-1)} \cdot s = 0 \pmod 2.$$

Solve by Gaussian elimination over $\mathrm{GF}(2)$ for the unique nonzero solution $s$.

**Why $n-1$?**

The equations $s \cdot z = 0$ leave a 1-dimensional solution space; together with $s \neq 0$ they pin down $s$ uniquely.

- Expected $O(n)$ **queries** to get enough independent equations.
- Post-processing is polynomial in $n$.

# Worked Example — Simon with $n = 2$, $s = 11$

## A valid oracle

Pair up $x$ and $x \oplus 11$: $\{00, 11\}$ and $\{01, 10\}$. Let

$$f(00) = f(11) = 00, \qquad f(01) = f(10) = 11.$$

## Solve

A useful outcome $z = 11$ gives $z_1 s_1 \oplus z_2 s_2 = s_1 \oplus s_2 = 0$, so $s_1 = s_2$. With $s \neq 00$ this forces $\boxed{s = 11}$. ✓

## One run

Suppose the output measures 00; input collapses to $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$. After $H^{\otimes 2}$ we see $z$ with $s \cdot z = 0$, i.e. $z \in \{00, 11\}$.

# Simon's Algorithm — Why It Matters

## The significance

- First natural problem with a **provable exponential** quantum–classical query gap (against *randomised* classical algorithms).
- Structure: quantum step produces *random linear constraints*; classical post-processing solves them.

## Bridge to Shor

Simon's "find the hidden period" is a *hidden-subgroup problem* over $(\mathbb{Z}_2)^n$. Shor's factoring solves the same template over $\mathbb{Z}$ using the *quantum Fourier transform* — our next unit.

# Quick Check — Round 3 (Simon's Algorithm)

**[Q1]**

State Simon's promise on $f$ and what the algorithm must find.

**[Q2]**

After one round, what condition does the measured string $z$ satisfy?

**[Q3]**

How many rounds (roughly) and what post-processing recover $s$?

**[Q4] MCQ**

Simon's quantum speed-up over classical algorithms is:
(a) constant (b) polynomial (c) exponential (d) none

# Answers — Round 3

**[A1]**

$f$ is two-to-one with $f(x) = f(x \oplus s)$ for a hidden $s \neq 0^n$; the goal is to find $s$.

**[A2]**

$z$ is uniformly random subject to $s \cdot z = 0 \pmod 2$ — one linear equation in $s$.

**[A3]**

About $O(n)$ rounds to obtain $n-1$ independent equations, then Gaussian elimination over $\mathrm{GF}(2)$ to solve for $s$.

**[A4]**

**(c) exponential** — $O(n)$ quantum queries vs. $\Theta(2^{n/2})$ classical.

# Summary of Unit 4

## What we covered

1. **Probabilistic vs. quantum** — amplitudes can be negative and *interfere*; probabilities cannot.
2. **Oracle model** — the reversible $U_f |x\rangle |y\rangle = |x\rangle |y \oplus f(x)\rangle$; count queries.
3. **Phase kick-back** — $U_f |x\rangle |-\rangle = (-1)^{f(x)} |x\rangle |-\rangle$: bit-oracle $\to$ phase-oracle.
4. **Deutsch** — constant vs. balanced $(f(0) \oplus f(1))$ in one query.
5. **Deutsch–Jozsa** — all-zeros $\Leftrightarrow$ constant, via $\frac{1}{2^n} \sum_x (-1)^{f(x)}$; one query.
6. **Honesty** — D–J beats only *deterministic* classical algorithms.
7. **Simon** — hidden $s$ from random constraints $s \cdot z = 0$; $O(n)$ vs. $\Theta(2^{n/2})$.
8. **Bridge** — Simon is a hidden-subgroup problem; Shor generalises it (next unit).

**Prepare a superposition, query once, interfere, then measure.**

# Practice Questions (Exam Oriented)

## Short answer (2–5 marks)

1. Explain phase kick-back and derive $U_f |x\rangle |-\rangle = (-1)^{f(x)} |x\rangle |-\rangle$.
2. State Deutsch's problem and its classical vs. quantum query counts.
3. State the Deutsch–Jozsa promise and the measurement rule.
4. State Simon's promise and the condition satisfied by each measured $z$.
5. Why is Simon's speed-up "stronger" than Deutsch–Jozsa's?

## Long answer (8–10 marks)

6. Derive the Deutsch algorithm end-to-end and interpret each measurement outcome.
7. Derive the Deutsch–Jozsa all-zeros amplitude and analyse the constant/balanced cases.
8. Present Simon's algorithm; show the post-Hadamard state is supported on $\{z : s \cdot z = 0\}$.
9. For $n = 2$, $s = 11$, build a valid oracle and recover $s$ from the samples.
10. Compare probabilistic and quantum computation, emphasising the role of interference.

# Think & Explore (Take-Home)

## Mini task (by hand)

- Run Deutsch for $f(x) = x$ (balanced) and $f(x) = 0$ (constant); confirm the outcomes.
- Trace Deutsch–Jozsa for $n = 2$ with a balanced $f$ of your choice.
- Finish the Simon example: list all valid outcomes and equations.

## Reading & reflection

- Kaye–Laflamme–Mosca, Ch. 6.
- Nielsen & Chuang, §1.4 (Deutsch, D–J) and Ch. 6.
- Prepare for **Unit 5**: the quantum Fourier transform, phase estimation and Shor's algorithm.

## Coding (self-learning)

In `Qiskit`: implement Deutsch–Jozsa for $n = 3$ and Simon for $n = 2$; verify the statistics on the simulator.

## Reminder

Assignment 2 covers Units 3 & 4 — handwritten, step-by-step, submitted as a single PDF via the course page.

# References

## Textbooks (as per the course page)


1. P. Kaye, R. Laflamme & M. Mosca, *An Introduction to Quantum Computing*, Oxford Univ. Press (Ch. 6). **[Main text]**
2. M. A. Nielsen & I. L. Chuang, *Quantum Computation and Quantum Information*, Cambridge.


## Seminal papers


3. D. Deutsch, "Quantum theory, the Church–Turing principle and the universal quantum computer," *Proc. R. Soc. A*, 1985.
4. D. Deutsch & R. Jozsa, "Rapid solution of problems by quantum computation," *Proc. R. Soc. A*, 1992.
5. D. R. Simon, "On the power of quantum computation," *FOCS*, 1994 (SIAM J. Comput., 1997).
6. R. Cleve, A. Ekert, C. Macchiavello & M. Mosca, "Quantum algorithms revisited," *Proc. R. Soc. A*, 1998.


NPTEL: *Quantum Computing* · Course page: https://www.gcjana.in/courses/shardauniversity/2501/quantum-computing/

# Thank You

Any Questions?

*"A quantum algorithm wins not by trying every answer at once, but by making the wrong answers interfere away."*

— the recurring theme of Unit 4

Post your doubts on the course page →

https://www.gcjana.in/courses/shardauniversity/2501/quantum-computing/#Post-doubts