Quantum Computing

Unit 1: Introduction — Church–Turing, Circuit Model, Reversible & Quantum Computation

Dr. Gopal Chandra Jana

Dept. of CSE, Sharda University

Quantum Computing (CSE063) — Unit 1

July 26, 2026

Course Page:

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

Outline

Learning Outcomes of Unit 1

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

1. **Explain** what a *model of computation* is and state the (Strong) Church–Turing thesis.
2. **Describe** the classical *circuit model* and identify *universal* gate sets.
3. **Formulate** bits, states and gates in *linear-algebra* (vector / matrix) language.
4. **Explain** *reversible* computation, Landauer's principle and reversible gates.
5. **Contrast** classical and quantum information; define a *qubit*, superposition and measurement.
6. **Motivate** why quantum physics enables new computational power.

Mapping to the Syllabus

A. Computers & the Strong Church–Turing thesis, circuit model of computation. **B.** Linear-algebra formulation, reversible computation. **C.** Quantum physics and computation.

What is a "Model of Computation"?

Idea

A **model of computation** makes precise what it means to *compute*: the allowed **states**, the elementary **steps**, and how inputs map to outputs.

- **Turing machine** (1936): tape + head + finite control — the reference model of *computability*.
- Equivalent models: λ-*calculus*, *RAM machine*, *cellular automata*, *Boolean circuits*.
- All compute exactly the same class of functions — the *computable* functions.

Key point

Different hardware, **same computable functions**. Physics decides which model is *realistic*.

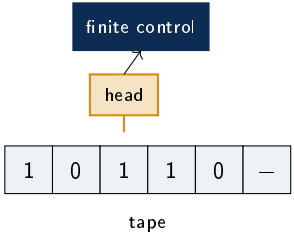


Figure: A Turing machine — the yardstick for what is *computable*.

The Church–Turing Thesis

Thesis (Church–Turing, 1936)

Every function that can be computed by an *effective / mechanical procedure* (an "algorithm") can be computed by a **Turing machine**.

- It is a **thesis**, not a theorem — it ties an *informal* notion ("algorithm") to a formal one.
- Cannot be *proved*, but no counter-example is known in $\sim$90 years.
- Draws the boundary of *computability*.

Consequence

Some problems are **undecidable** — no algorithm exists at all, e.g. the *Halting Problem* (will program P on input x ever stop?).

The Strong (Extended) Church–Turing Thesis

Strong Church–Turing Thesis

A **probabilistic Turing machine** can *efficiently* simulate any "realistic" model of computation — with only **polynomial** overhead in time.

- Ordinary C–T thesis: about **what** can be computed (computability).
- *Strong* version: about **how efficiently** — "efficient" = polynomial time.
- It elevates *polynomial-time* to a model-independent notion of tractability.

Why it matters for us

Quantum computers appear to **violate** the Strong C–T thesis: Shor's algorithm factors integers in polynomial time, with *no* known efficient classical counterpart. **This is the doorway to quantum computing.**

A Glance at Complexity Classes

Efficient = polynomial time

- **P** — solvable in poly-time (deterministic).
- **NP** — solutions *verifiable* in poly-time.
- **BPP** — bounded-error *probabilistic* poly-time (the "efficient classical" class).
- **BQP** — bounded-error *quantum* poly-time.

Belief

$\mathsf{P} \subseteq \mathsf{BPP} \subseteq \mathsf{BQP}$, and problems such as *factoring* are believed to lie in $\mathsf{BQP} \setminus \mathsf{BPP}$.

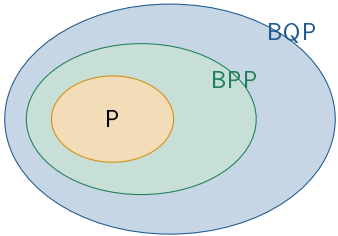


Figure: Conjectured inclusions of efficient classes.

The Circuit Model of Computation

Definition

A **Boolean circuit** is an *acyclic* network of **gates** acting on **wires** that carry bits, computing a function $f : \{0,1\}^n \to \{0,1\}^m$.

- Elementary gates: **NOT, AND, OR, XOR, NAND**, plus **FANOUT** (copy a bit).
- A *circuit family* $\{C_n\}$ has one circuit per input size n.
- Cost measures: **size** (# gates) and **depth** (longest path).

Equivalence

Uniform circuit families are equivalent in power to Turing machines.

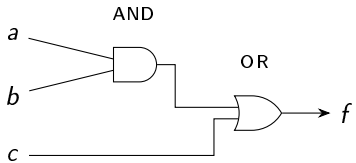


Figure: A small circuit $f = (a \wedge b) \vee c$.

Universal Gate Sets

Universality

A gate set is **universal** if *any* Boolean function can be built from it (with FANOUT).

- $\{\text{AND}, \text{OR}, \text{NOT}\}$ is universal.
- $\{\text{NAND}\}$ *alone* is universal — and so is $\{\text{NOR}\}$.
- FANOUT lets a wire's value be reused.

NAND builds everything

$$\text{NOT}(a) = a \mathbin{\overline{\wedge}} a$$
$$\text{AND}(a, b) = \text{NOT}(a \mathbin{\overline{\wedge}} b)$$
$$\text{OR}(a, b) = (a \mathbin{\overline{\wedge}} a) \mathbin{\overline{\wedge}} (b \mathbin{\overline{\wedge}} b)$$

(here $\mathbin{\overline{\wedge}}$ denotes NAND).

Worked Example — The Half Adder

Task

Add two bits a, b. Outputs: **sum** $s = a \oplus b$ and **carry** $c = a \wedge b$.

Truth table

a	b	s	c
0	0	0	0
0	1	1	0
1	0	1	0
1	1	**0**	**1**

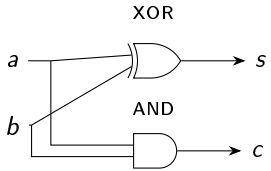


Figure: Half adder = one XOR (sum) + one AND (carry).

Quick Check — Round 1 (Church–Turing & Circuits)

[Q1]

In one line, what does the Church–Turing thesis assert?

[Q2]

How does the *Strong* Church–Turing thesis differ from the ordinary one?

[Q3]

Is $\{\text{NAND}\}$ a universal gate set? Justify briefly.

[Q4] True/False

"The circuit model can compute strictly more functions than a Turing machine."

Think for a minute — answers on the next slide.

Answers — Round 1

[A1]

Anything computable by an effective / algorithmic procedure can be computed by a **Turing machine**.

[A2]

C–T thesis is about **computability** (*what* can be computed); the Strong version is about **efficiency** — any realistic model is simulable with *polynomial* overhead.

[A3]

Yes. NAND builds NOT, AND and OR, hence every Boolean function; it is *functionally complete*.

[A4]

False. Uniform circuit families and Turing machines compute exactly the *same* class of functions.

A Linear-Algebra View — Bits as Vectors

Encoding

$$0 \longleftrightarrow |0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \qquad 1 \longleftrightarrow |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}.$$

A bit-string is a **standard basis vector**. For n bits the state lives in a space of dimension 2^n.

- **Deterministic** state = a basis vector (a single "1").
- **Probabilistic** state = a *probability vector* (entries ≥ 0, summing to 1).

Two bits, four basis states

$$|00\rangle, |01\rangle, |10\rangle, |11\rangle \in \mathbb{R}^4.$$

e.g. $|10\rangle = (0, 0, 1, 0)^\top$.

Reading

Computation becomes **matrix–vector multiplication** on these vectors.

Gates as Matrices

The rule

A gate on n bits is a $2^n \times 2^n$ matrix M; applying it is

$$|\text{out}\rangle = M\,|\text{in}\rangle\,.$$

- *Reversible / deterministic* gate $\Rightarrow$ **permutation matrix**.
- *Probabilistic* gate $\Rightarrow$ **stochastic matrix**.

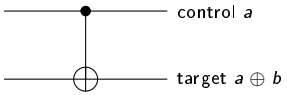
NOT gate

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad X\,|0\rangle = |1\rangle\,,\; X\,|1\rangle = |0\rangle\,.$$

CNOT (controlled-NOT)

Two bits: $|a, b\rangle \mapsto |a,\, a \oplus b\rangle$.

$$\text{CNOT} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$



Composing Systems — The Tensor Product

Joining independent systems

The joint state of separate systems is their **tensor product** $\otimes$:

$$|0\rangle \otimes |1\rangle = \begin{pmatrix}1\\0\end{pmatrix} \otimes \begin{pmatrix}0\\1\end{pmatrix} = \begin{pmatrix}0\\1\\0\\0\end{pmatrix} = |01\rangle\,.$$

- Dimensions **multiply**: $2^m \otimes 2^n = 2^{m+n}$.
- Gates on disjoint wires combine as $M_1 \otimes M_2$.

Why it matters

n bits $\Rightarrow$ a vector of length 2^n. This **exponential** state space is exactly what quantum computing exploits (soon).

Note

Tensor product $\neq$ ordinary vector addition — it is how *sub-systems* are stacked.

Reversible Computation — Why Bother?

Landauer's principle (1961)

Erasing one bit of information must dissipate at least

$$E_{\min} = k_B\, T \ln 2$$

of heat. *Logical* irreversibility $\Rightarrow$ *physical* energy cost.

- AND, OR are **irreversible**: from the output you cannot recover the inputs $\Rightarrow$ information (and energy) lost.
- **Bennett (1973):** any computation can be made *reversible* $\Rightarrow$ in principle *zero* dissipation.

Bridge to quantum

Quantum gates are unitary $\Rightarrow$ reversible. So we first learn *reversible classical* logic — it is the classical shadow of quantum gates.

Reversible Gates

The toolbox

- **NOT** — 1 bit, reversible.
- **CNOT** — 2 bits: $|a, b\rangle \mapsto |a, a \oplus b\rangle$.
- **Toffoli (CCNOT)** — 3 bits: $|a, b, c\rangle \mapsto |a, b, c \oplus (a \wedge b)\rangle$.
- **Fredkin (CSWAP)** — controlled swap.

Universality

Toffoli is universal for reversible classical computation (and, with ancilla/garbage bits, simulates any circuit).

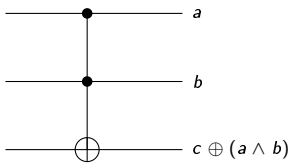


Figure: The Toffoli gate — two controls, one target.

Worked Example — The Toffoli Gate

Truth table $|a, b, c\rangle \mapsto |a, b, c \oplus ab\rangle$

a	b	c	a'	b'	c'
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	0
0	1	1	0	1	1
1	0	0	1	0	0
1	0	1	1	0	1
1	1	0	1	1	**1**
1	1	1	1	1	**0**

It computes AND

Set the target $c = 0$:

$$|a, b, 0\rangle \;\mapsto\; |a, b,\ a \wedge b\rangle\,.$$

The third bit is $a \wedge b$ — AND, done *reversibly*.

Hence universal

With $a = b$ we get $a \oplus a \cdot 1$ tricks, and combining copies of Toffoli reproduces NAND, so every Boolean function follows.

Quick Check — Round 2 (Linear Algebra & Reversibility)

[Q1]

Write $|0\rangle$ and $|1\rangle$ as column vectors and give the matrix of the NOT gate.

[Q2]

Why is AND irreversible, and what does Landauer's principle predict as a consequence?

[Q3]

State the Toffoli mapping and explain why it is universal for reversible computing.

[Q4]

Compute $|1\rangle \otimes |0\rangle$ and state the dimension of the resulting space.

Answers — Round 2

[A1]

$|0\rangle = (1,0)^\top$, $|1\rangle = (0,1)^\top$; $\quad X = \left(\begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix}\right)$, so $X|0\rangle = |1\rangle$.

[A2]

AND maps $(1,0), (0,1), (0,0)$ all to 0: the inputs cannot be recovered $\Rightarrow$ information erased. Landauer $\Rightarrow$ at least $k_B T \ln 2$ of **heat** per erased bit.

[A3]

$|a, b, c\rangle \mapsto |a, b,\ c \oplus (a \wedge b)\rangle$. Setting $c = 0$ yields AND (and thus NAND), which is functionally complete — so any Boolean function is realisable.

[A4]

$|1\rangle \otimes |0\rangle = |10\rangle = (0,0,1,0)^\top$, a vector in a space of dimension $2^2 = 4$.

A Preview of Quantum Physics

Four ideas we will use

- **Superposition:** a system can be in a *linear combination* of basis states.
- **Amplitudes:** complex numbers α; probability $= |\alpha|^2$ (*Born rule*).
- **Measurement:** probabilistic, and *disturbs* the state (collapse).
- **Entanglement:** correlations with *no* classical analogue.

From probabilistic to quantum

Probabilistic bit: state = probability vector, entries ≥ 0 summing to 1.

Qubit: state = **amplitude** vector, complex entries with $\sum |\alpha_i|^2 = 1$. Amplitudes can be *negative / complex* $\Rightarrow$ **interference**.

No-cloning

An *unknown* quantum state cannot be copied — unlike a classical FANOUT.

The Qubit

State of one qubit

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle\,, \quad \alpha, \beta \in \mathbb{C}, \quad |\alpha|^2 + |\beta|^2 = 1.$$

- A classical bit is a *point* in $\{0, 1\}$.
- A qubit is a *unit vector* in $\mathbb{C}^2$ — visualised on the **Bloch sphere**.
- n qubits $\Rightarrow$ a unit vector in $\mathbb{C}^{2^n}$ (2^n amplitudes).

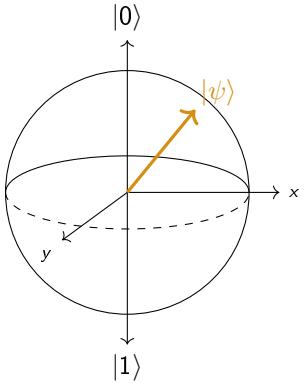


Figure: A qubit on the Bloch sphere.

Measurement & the Born Rule

Measuring in the computational basis

For $|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$:

$$\Pr[0] = |\alpha|^2, \qquad \Pr[1] = |\beta|^2.$$

The state then **collapses** to the observed basis state.

- Measurement is **probabilistic** and **irreversible**.
- You get *one* classical bit out, not the amplitudes.

Example

$|\psi\rangle = \frac{\sqrt{3}}{2}\,|0\rangle + \frac{1}{2}\,|1\rangle$:

$$\Pr[0] = \tfrac{3}{4}, \qquad \Pr[1] = \tfrac{1}{4}.$$

Check: $\frac{3}{4} + \frac{1}{4} = 1$. ✓

Consequence

Extracting information *destroys* superposition — good algorithm design must arrange **constructive interference** on the right answer *before* measuring.

Quantum Gates

Gates are unitary

A quantum gate is a **unitary** matrix U with $U^\dagger U = I$ (so $U^{-1} = U^\dagger$ — always reversible).

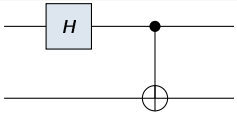
$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix},\ Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix},\ H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

Hadamard makes superposition

$$H|0\rangle = \frac{1}{\sqrt{2}}\big(|0\rangle + |1\rangle\big).$$

Universal quantum gate sets

e.g. $\{H, T, \text{CNOT}\}$ can approximate any unitary to arbitrary precision.



creates entanglement (Bell state)

Figure: H then CNOT $\Rightarrow$ an entangled pair.

Quantum vs. Classical Computation — At a Glance

Aspect	Classical	Quantum
State of a unit	bit $\in \{0,1\}$	qubit: unit vector in $\mathbb{C}^2$
n units	string in $\{0,1\}^n$	unit vector in $\mathbb{C}^{2^n}$
Composition	direct / Cartesian	**tensor product** $\otimes$
Evolution	Boolean / stochastic maps	**unitary** maps $U^\dagger U = I$
Read-out	deterministic	**probabilistic** measurement (Born rule)
Copying a state	allowed (FANOUT)	**forbidden** (no-cloning)

One-line memory hook

Classical computing moves *bits*; quantum computing steers *amplitudes* — and lets them **interfere**.

Quantum Physics and Computation — Why It's Powerful

The resource

n qubits carry 2^n complex amplitudes evolving *in parallel*. Superposition + entanglement + **interference** let the right answers reinforce and the wrong ones cancel.

- **Feynman (1982):** simulating quantum systems is hard classically — so *use* a quantum machine.
- **Deutsch (1985):** a *universal quantum computer* — and a direct challenge to the Strong C–T thesis.
- **Algorithms:** Shor (factoring), Grover (search), Hamiltonian simulation.

Reality check

A large state space is *not* free lunch — measurement returns only one string. The art is **interference**: engineer amplitudes so the desired outcome is likely.

Where we're headed

Unit 2+ : single-qubit gates, Bloch geometry, multi-qubit circuits, and the first quantum algorithms.

Quick Check — Round 3 (Quantum Basics)

[Q1]

Write the general one-qubit state and its normalisation condition.

[Q2]

State the Born rule. For $|\psi\rangle = \frac{\sqrt{3}}{2}|0\rangle + \frac{1}{2}|1\rangle$, find Pr[1].

[Q3]

Which mathematical property guarantees that every quantum gate is reversible?

[Q4] MCQ

Which is *not* a feature of quantum information?
(a) superposition (b) entanglement (c) cloning an unknown state (d) interference

Answers — Round 3

[A1]

$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$ with $\alpha, \beta \in \mathbb{C}$ and $|\alpha|^2 + |\beta|^2 = 1$.

[A2]

Born rule: Pr[outcome k] $= |\text{amplitude}_k|^2$. Here Pr[1] $= \left(\frac{1}{2}\right)^2 = \mathbf{0.25}$.

[A3]

Unitarity ($U^\dagger U = I$): every unitary has an inverse $U^{-1} = U^\dagger$.

[A4]

(c) cloning an unknown state — forbidden by the *no-cloning theorem*.

Summary of Unit 1

What we covered

1. **Models of computation** — Turing machine as the yardstick of computability.
2. **Church–Turing thesis** — and its *Strong* form about polynomial-time efficiency.
3. **Circuit model** — gates, universal sets (NAND), size and depth.
4. **Linear-algebra view** — bits as vectors, gates as matrices, tensor products.
5. **Reversible computation** — Landauer's principle, CNOT, Toffoli, Fredkin.
6. **Quantum physics** — superposition, amplitudes, Born-rule measurement, no-cloning.
7. **The qubit** — unit vectors in $\mathbb{C}^{2^n}$; unitary gates (H, X, Z, CNOT).
8. **Why quantum** — interference and entanglement challenge the Strong C–T thesis.

Reversible logic is the classical bridge to unitary quantum computation.

Practice Questions (Exam Oriented)

Short answer (2–5 marks)

1. State the Church–Turing thesis and distinguish it from its Strong version.
2. Show that $\{\text{NAND}\}$ is a universal gate set.
3. Represent $|0\rangle, |1\rangle$ as vectors and write the NOT and CNOT matrices.
4. State Landauer's principle and explain why AND is irreversible.
5. Give the general qubit state and the Born rule with one example.

Long answer (8–10 marks)

6. Explain the circuit model; design and analyse a half adder.
7. Describe the Toffoli gate; prove it is universal for reversible computation.
8. Contrast classical and quantum computation across state, evolution and read-out.
9. Explain superposition, measurement and no-cloning; why can't we just read all 2^n amplitudes?
10. Discuss how quantum computing challenges the Strong Church–Turing thesis, with an example.

Think & Explore (Take-Home)

Mini design task

On paper, build a **reversible 1-bit full adder**:

- Inputs: a, b, c_{in} (plus ancilla).
- Use only NOT / CNOT / Toffoli.
- Identify the "garbage" bits and the useful outputs.

Coding (self-learning)

Implement H then CNOT on two qubits in `Qiskit`, then measure many shots and plot the histogram — observe the Bell-state correlations.

Reading & reflection

- Kaye, Laflamme & Mosca — Ch. 1 (background & reversibility).
- Nielsen & Chuang — Ch. 1–2 (intro & linear algebra).
- Prepare for **Unit 2**: single-qubit gates and the Bloch sphere.

Reminder

Assignment 1 covers Unit 1 — 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. **[Main text]**
2. M. A. Nielsen & I. L. Chuang, *Quantum Computation and Quantum Information*, Cambridge.
3. N. D. Mermin, *Quantum Computer Science: An Introduction*, Cambridge.

Seminal papers

4. R. Landauer, "Irreversibility and heat generation in the computing process," *IBM J. R&D*, 1961.
5. R. P. Feynman, "Simulating physics with computers," *Int. J. Theor. Phys.*, 1982.
6. D. Deutsch, "Quantum theory, the Church–Turing principle and the universal quantum computer," *Proc. R. Soc. A*, 1985.
7. P. W. Shor, "Algorithms for quantum computation: discrete logarithms and factoring," *FOCS*, 1994.

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

Thank You

Any Questions?

"Nature isn't classical, dammit, and if you want to make a simulation of nature, you'd better make it quantum mechanical."

— Richard P. Feynman (1982)

Post your doubts on the course page →

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