Why $\mathbf{P} \subseteq \mathbf{BQP} \subseteq \mathbf{PP} \subseteq \mathbf{PSPACE}$

and the conceptually clean route via $\mathbf{PostBQP} = \mathbf{PP}$

Goal

We prove (as in Nielsen & Chuang) that

$$\boxed{P \subseteq BQP.}$$

Core steps:

1. Any deterministic polytime classical computation can be implemented by a *polynomial-size reversible circuit*.
2. Reversible circuits can be built from *Toffoli* (plus NOT/CNOT) gates.
3. Each Toffoli gate is a unitary operator on qubits, hence the whole reversible circuit is a unitary U.

Therefore a quantum computer can simulate any P computation with polynomial overhead.

BQP model needed here

To show $P \subseteq BQP$, it suffices to show:
Given a deterministic polytime TM (or Boolean circuit family) computing a decision function

$$f : \{0,1\}^n \to \{0,1\},$$

there is a polynomial-size quantum circuit that outputs $f(x)$ with probability 1.
We'll construct a unitary U_f such that

$$U_f : \ |x\rangle|0\rangle \mapsto |x\rangle|f(x)\rangle$$

(possibly using ancillas, later "uncomputed").

Why reversibility matters

A quantum circuit must be unitary, hence *reversible*.
But classical logic gates like AND, OR are *irreversible*:

$$(x, y) \mapsto x \wedge y \quad \text{loses information about } (x, y).$$

Standard trick: compute $f(x)$ into an extra register without destroying x:

$$(x, b) \mapsto (x, b \oplus f(x)).$$

This is reversible because given $(x, b \oplus f(x))$ we can recover b.
So we want a reversible circuit for the map:

$$|x\rangle|b\rangle \mapsto |x\rangle|b \oplus f(x)\rangle.$$

Toffoli gate (CCNOT)

The Toffoli gate acts on 3 bits/qubits:

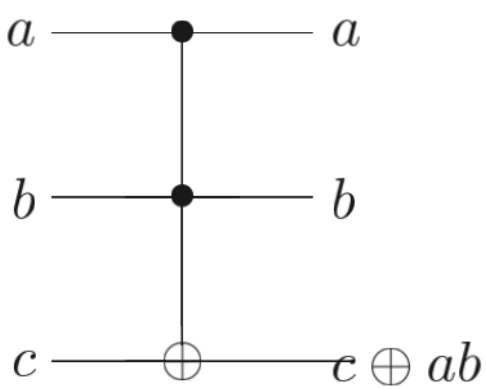
$$\text{Toffoli}: \ (a, b, c) \mapsto (a, b, c \oplus (a \wedge b)).$$

Truth-table effect: flip the target c iff both controls are 1.
As a quantum gate, Toffoli is a permutation of computational basis states, hence represented by an 8×8 *unitary* matrix (a permutation matrix).
Key fact (classical reversible universality):

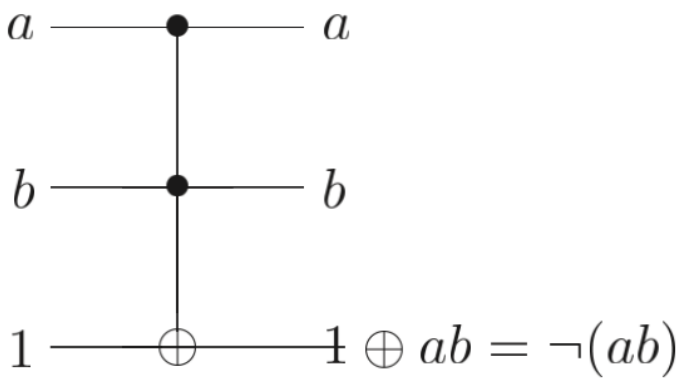
Toffoli gates generate all reversible Boolean functions.

Reversible realizations of AND, NAND, and FANOUT

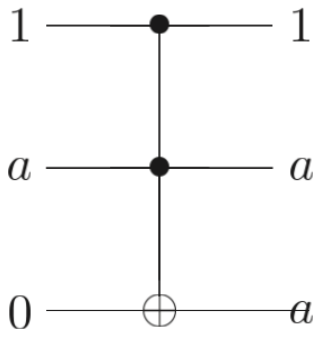
AND



NAND



FANOUT



- AND: Toffoli with clean target.
- NAND: AND followed by NOT.
- FANOUT: CNOT copies a bit into an ancilla.

Toffoli as a unitary matrix

In the computational basis ordered as

$$|000\rangle, |001\rangle, |010\rangle, |011\rangle, |100\rangle, |101\rangle, |110\rangle, |111\rangle,$$

Toffoli acts as identity except it swaps $|110\rangle \leftrightarrow |111\rangle$.
Equivalently, its matrix is the identity with the bottom-right 2×2 block replaced by

$$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}.$$

Thus Toffoli is unitary and implementable in a quantum circuit model.

Embedding an arbitrary classical computation

Let a classical polytime computation (TM or circuit) compute $f(x)$.
Nielsen & Chuang route (circuit model):

- Any polytime TM can be compiled into a polynomial-size Boolean circuit family $\{C_n\}$.
- We then convert C_n (typically with AND/OR/NOT) into a polynomial-size *reversible* circuit R_n.

So it suffices to show:

Any Boolean circuit C can be made reversible with poly overhead.

Reversible simulation of irreversible gates: the standard trick

Irreversible gate $g(u, v) = u \wedge v$ can be simulated reversibly by computing into an ancilla:

$$(u, v, 0) \mapsto (u, v, u \wedge v).$$

This is exactly a Toffoli gate with target initialized to 0:

$$(a, b, c) \mapsto (a, b, c \oplus (a \wedge b)), \quad c = 0.$$

Similarly:

- NOT: reversible by itself.
- XOR: reversible by itself (CNOT).
- FANOUT: not reversible as a standalone map $x \mapsto (x, x)$, but can be done reversibly using CNOT:

$$(x, 0) \mapsto (x, x).$$

Thus we can build reversible versions of a standard classical gate set.

From a Boolean circuit to a reversible circuit

Let C be a size-s Boolean circuit computing $f(x)$.
Compute each gate's output into a fresh ancilla wire, leaving inputs intact.
This yields a reversible circuit R that maps

$$|x\rangle|0^m\rangle|0\rangle \mapsto |x\rangle|g(x)\rangle|f(x)\rangle,$$

where $g(x)$ is "garbage" (all intermediate gate values) and $m = O(s)$.
Overhead so far:

- Ancillas: $m = O(s)$
- Gates: $O(s)$ (each AND becomes one Toffoli; wiring via CNOT/NOT)

Still polynomial if $s = \mathrm{poly}(n)$.

Cleaning up garbage: Bennett's uncomputation

We want a clean output of the form $|x\rangle|f(x)\rangle$ (plus zeros), not extra junk. Bennett trick (standard in N&C):

1. Compute forward to get $|x\rangle|g(x)\rangle|f(x)\rangle$.
2. Copy out $f(x)$ to a fresh output bit by CNOT:

$$|f(x)\rangle|0\rangle \mapsto |f(x)\rangle|f(x)\rangle.$$

3. Run the computation *backwards* to uncompute garbage:

$$|x\rangle|g(x)\rangle|f(x)\rangle \mapsto |x\rangle|0^m\rangle|0\rangle.$$

Net effect:

$$|x\rangle|0^m\rangle|0\rangle|0\rangle \mapsto |x\rangle|0^m\rangle|0\rangle|f(x)\rangle.$$

Polynomial overhead bound

If the original circuit has size $s = \mathrm{poly}(n)$:

- Forward reversible computation: $O(s)$ Toffoli/CNOT/NOT gates, $O(s)$ ancillas.
- Copy-out: $O(1)$ (one CNOT).
- Reverse computation: another $O(s)$ gates.

Total gate count:

$$O(s) + O(1) + O(s) = O(s) = \mathrm{poly}(n).$$

Total ancillas:

$$O(s) = \mathrm{poly}(n).$$

Thus we obtain a poly-size quantum circuit implementing a unitary U_f.

Putting it together: $P \subseteq BQP$

Let $L \in P$. Then there exists a polytime deterministic algorithm computing $f_L(x) \in \{0,1\}$.
By the construction:

- Compile the classical computation to a poly-size Boolean circuit.
- Convert it to a poly-size reversible circuit using Toffoli gates and ancillas.
- Interpret each reversible gate as a quantum unitary.

We obtain a quantum circuit that outputs $f_L(x)$ with probability 1. Hence:

$$\boxed{L \in BQP.} \quad \Rightarrow \quad \boxed{P \subseteq BQP.}$$

Optional: decision vs function form

Decision version (language L):

$$|x\rangle|0\rangle \mapsto |x\rangle|f_L(x)\rangle, \quad \text{measure last qubit.}$$

Function version (general $f : \{0,1\}^n \to \{0,1\}^t$): repeat the output-register trick bitwise:

$$|x\rangle|0^t\rangle \mapsto |x\rangle|f(x)\rangle$$

with polynomial overhead in circuit size and output length.

Takeaways

- Quantum circuits are reversible (unitary), so we simulate classical computation via reversible computation.
- Toffoli is universal for reversible classical computing and is itself a unitary.
- Garbage can be removed by uncomputation with only a constant-factor overhead.

Therefore:

$$\boxed{P \subseteq BQP.}$$

BQP $\subseteq$ PSPACE: Scope

We focus only on the classes needed for:

$$\boxed{\text{BQP} \subseteq \text{PSPACE}}$$

and provide a conceptual proof of:

$$\boxed{\text{PostBQP} = \text{PP}} \quad \text{(Aaronson)}$$

Then we get:

$$\text{BQP} \subseteq \text{PostBQP} = \text{PP} \subseteq \text{PSPACE}.$$

Definitions: BQP and PP

BQP (bounded-error quantum polytime). A language L is in BQP if there exists a uniform poly-size quantum circuit family such that

$$x \in L \Rightarrow \Pr[\text{accept}] \geq 2/3, \qquad x \notin L \Rightarrow \Pr[\text{accept}] \leq 1/3.$$

Amplification reduces error exponentially.

PP (probabilistic polytime with majority threshold). $L \in PP$ iff there exists a probabilistic polytime TM M such that

$$x \in L \Rightarrow \Pr[M(x) = 1] > 1/2, \qquad x \notin L \Rightarrow \Pr[M(x) = 1] \leq 1/2.$$

No promise on the gap from $1/2$.

Definition: PostBQP

PostBQP is BQP augmented with *postselection*:

- The circuit produces two designated output bits (p, a):
  - $p =$ "postselection" bit
  - $a =$ "answer" bit
- We are allowed to *condition* on the event $p = 1$ (even if $\Pr[p = 1]$ is exponentially small),
- and decide based on $\Pr[a = 1 \mid p = 1]$ with bounded error:

  $$x \in L \Rightarrow \Pr[a = 1 \mid p = 1] \geq 2/3, \qquad x \notin L \Rightarrow \Pr[a = 1 \mid p = 1] \leq 1/3,$$

  with the promise $\Pr[p = 1] > 0$.

Intuition: postselection lets us "throw away" all runs except a rare event.

High-level roadmap

We will show:

$$\boxed{\text{PostBQP} = \text{PP}}$$

via two inclusions:

$$\text{PostBQP} \subseteq \text{PP} \quad \text{and} \quad \text{PP} \subseteq \text{PostBQP}.$$

Then:

$$\boxed{\text{BQP} \subseteq \text{PP} \subseteq \text{PSPACE}}$$

follows immediately, since $\text{BQP} \subseteq \text{PostBQP}$.

Key algebra: postselection turns ratios into decisions

Postselection computes a *conditional probability*:

$$\Pr[a = 1 \mid p = 1] = \frac{\Pr[a = 1 \wedge p = 1]}{\Pr[p = 1]}.$$

Thus, if we can engineer quantum circuits so that

$$\Pr[a = 1 \wedge p = 1] \propto A(x), \qquad \Pr[p = 1] \propto B(x),$$

then postselection gives access to the ratio $A(x)/B(x)$.
Aaronson's insight: PP is essentially about comparing certain exponentially-small differences, and postselection gives exactly that "amplification" power.

Inclusion 1: PostBQP $\subseteq$ PP (idea)

Take a PostBQP circuit C_x with output bits (p, a).
We need to decide whether

$$\Pr[a = 1 \mid p = 1] > 1/2 \quad \text{(after amplification to make the gap robust).}$$

Equivalently:

$$\Pr[a = 1 \wedge p = 1] - \Pr[a = 0 \wedge p = 1] \; > \; 0.$$

So it suffices to show that *the sign of a certain difference of probabilities* from a quantum circuit is decidable in PP.
This reduces to: PP can decide whether a *gap* of two path-sums is positive.

PostBQP $\subseteq$ PP: path-sum / GapP sketch

Fix a standard finite universal gate set (e.g., Hadamard + Toffoli), so amplitudes lie in

$$\left\{ \frac{m}{2^{k/2}} : m \in \mathbb{Z},\ k = \mathrm{poly}(n) \right\}.$$

Each amplitude is a sum over exponentially many computational paths with $\pm$ contributions. Hence each probability such as $\Pr[a = b \wedge p = 1]$ can be written as

$$\Pr[a = b \wedge p = 1] = \frac{g_b(x)}{2^{q(n)}}$$

for an integer-valued *gap function* $g_b(x)$ (difference of two #counts). Therefore the quantity

$$\Pr[a = 1 \wedge p = 1] - \Pr[a = 0 \wedge p = 1] = \frac{g_1(x) - g_0(x)}{2^{q(n)}}$$

has sign equal to the sign of an integer gap, which PP can decide. Thus:

$$\boxed{\mathrm{PostBQP} \subseteq \mathrm{PP}.}$$

Inclusion 2: PP $\subseteq$ PostBQP (conceptual core)

Let $L \in PP$. Then there exists a polytime probabilistic TM M such that:

$$x \in L \Rightarrow \Pr[M(x) = 1] > 1/2, \qquad x \notin L \Rightarrow \Pr[M(x) = 1] \leq 1/2.$$

Let $p(x) = \Pr[M(x) = 1]$. We want a postselected quantum circuit that decides whether $p(x) > 1/2$.
Write M as a polytime computation using m random bits $r \in \{0,1\}^m$. Let $f_x(r) \in \{0,1\}$ be the accept indicator. Then:

$$p(x) = \frac{1}{2^m} \sum_r f_x(r).$$

So PP is about whether

$$\sum_r f_x(r) \; > \; 2^{m-1}.$$

PP $\subseteq$ PostBQP: build a bias into an amplitude

Prepare uniform superposition over random strings:

$$\frac{1}{\sqrt{2^m}} \sum_{r \in \{0,1\}^m} |r\rangle |f_x(r)\rangle.$$

Now apply a Hadamard on the $|f_x(r)\rangle$ qubit (call it the "flag" qubit). Conditioned on the flag being $|1\rangle$, the amplitude of the all-zero $|0^m\rangle$ state encodes a quantity proportional to:

$$\sum_r (-1)^{f_x(r)} = \#\{r : f_x(r) = 0\} - \#\{r : f_x(r) = 1\}.$$

Notice:

$$\sum_r (-1)^{f_x(r)} < 0 \iff \#\text{accept} > \#\text{reject} \iff p(x) > 1/2.$$

So the *sign* of a computable amplitude distinguishes $x \in L$ vs $x \notin L$. Postselection will be used to convert that tiny amplitude difference into a constant gap.

PP $\subseteq$ PostBQP: postselection amplifies exponentially small bias

The amplitude gap distinguishing cases can be exponentially small, e.g.

$$p(x) = 1/2 + 2^{-m}.$$

A normal BQP machine cannot reliably resolve it.
Postselection allows conditioning on a rare event so that the *conditional* probability becomes bounded away from $1/2$.
Concretely, Aaronson shows how to engineer a postselection event $p = 1$ such that

$$\Pr[a = 1 \mid p = 1] = \frac{1}{2}\left(1 + \frac{\sum_r (-1)^{f_x(r)}}{2^m}\right),$$

and then apply standard amplification within PostBQP to separate the cases:

$$\sum_r (-1)^{f_x(r)} < 0 \Rightarrow \Pr[a = 1 \mid p = 1] \geq 2/3,$$

$$\sum_r (-1)^{f_x(r)} \geq 0 \Rightarrow \Pr[a = 1 \mid p = 1] \leq 1/3.$$

Conclusion: PostBQP = PP

We have both containments:

$$\text{PostBQP} \subseteq \text{PP} \quad \text{and} \quad \text{PP} \subseteq \text{PostBQP}.$$

Therefore:

$$\boxed{\text{PostBQP} = \text{PP}.}$$

Interpretation:

- Postselection gives exactly PP-level power.
- PP captures the ability to resolve exponentially tiny biases.

Deriving BQP $\subseteq$ PP cleanly

Since BQP is just PostBQP without using postselection:

$$\boxed{\text{BQP} \subseteq \text{PostBQP} = \text{PP}.}$$

This route is conceptually cleaner than the original path-sum proof, because it isolates "what extra power is needed" (postselection).

Why PP $\subseteq$ PSPACE (sketch)

A PP machine M can be viewed as a polytime computation tree with 2^m leaves.
Define $A(x) =$ number of accepting leaves.
PP asks whether:

$$A(x) > 2^{m-1}.$$

Compute $A(x)$ using depth-first recursion:

- Recurse on partial random strings
- Accumulate counts on the call stack
- Maintain only $O(m)$ bits of recursion depth and counters of size $O(m)$ bits

Hence the counting is doable in polynomial space (time may be exponential):

$$\boxed{\text{PP} \subseteq \text{PSPACE}.}$$

Final containment: BQP $\subseteq$ PSPACE

Putting it together:

$$\mathrm{BQP} \subseteq \mathrm{PostBQP} = \mathrm{PP} \subseteq \mathrm{PSPACE}.$$

Therefore:

$$\boxed{\mathrm{BQP} \subseteq \mathrm{PSPACE}.}$$

Meaning:

- Any quantum polytime computation can be simulated with polynomial space.
- No claim of polynomial *time* simulation.

Additional notes

- The PostBQP = PP proof is due to Scott Aaronson ("Quantum Computing, Postselection, and Probabilistic Polynomial-Time", 2005).
- For a fully formal proof, one typically:
  - fixes a gate set (e.g., Hadamard+Toffoli),
  - uses exact rational / dyadic amplitude representations,
  - expresses probabilities as GapP$/2^{poly}$,
  - and uses postselection gadgets to convert amplitude sign into bounded-error conditional acceptance.
- If desired, one can add a worked example with a tiny m showing the amplitude sign flip.