Introduction to Quantum Computing

Grover and Shor's Algorithms

Ji, YongHyeon

Department of Cyber Security
Kookmin University

April 24, 2025

Contents

Chapter 1

Linear Operators on Finite-Dimensional Hilbert Spaces

Quantum computing relies fundamentally on the language of linear algebra. Quantum states are vectors in complex Hilbert spaces, and quantum operations are linear operators acting on these spaces.

1.1 Vector Spaces and Dirac's Bra–Ket Notation

Definition 1.1 (Hilbert Space). A **Hilbert space** $\mathcal{H}$ is a complete inner product space over the field $\mathbb{F}$ (either $\mathbb{R}$ or $\mathbb{C}$). Concretely, $\mathcal{H}$ satisfies:

1. (Vector Space) $\mathcal{H}$ is a vector space over $\mathbb{F}$.
2. (Inner Product) There exists a map

$$\langle \cdot, \cdot \rangle : \mathcal{H} \times \mathcal{H} \to \mathbb{F}$$

   satisfying for all $x, y, z \in \mathcal{H}$ and $\alpha, \beta \in \mathbb{F}$:

   (a) (Conjugate Symmetry) $\langle x, y \rangle = \overline{\langle y, x \rangle}$.
   (b) (Linearity) $\langle \alpha x + \beta y,\, z \rangle = \alpha \langle x, z \rangle + \beta \langle y, z \rangle$.
   (c) (Positive-Definiteness) $\langle x, x \rangle \geq 0$, and $\langle x, x \rangle = 0$ if and only if $x = 0$.

3. (Norm) The norm induced by the inner product, $\|x\| = \sqrt{\langle x, x \rangle}$, defines a metric $d(x, y) = \|x - y\|$.
4. (Completeness) $\mathcal{H}$ is complete with respect to the metric d, i.e., every Cauchy sequence in $\mathcal{H}$ converges to a limit in $\mathcal{H}$.

Definition 1.2 (Ket). Let $\mathcal{H}$ be a Hilbert space over $\mathbb{C}$. A **ket** $|\psi\rangle$ denotes an element $\psi \in \mathcal{H}$.

Definition 1.3 (Bra). To each ket $|\psi\rangle \in \mathcal{H}$, there corresponds a unique continuous linear functional (via the Riesz representation theorem) denoted by the **bra** $\langle\psi| : \mathcal{H} \to \mathbb{C}$, defined by

$$\langle\psi| \left(|\phi\rangle \right) = \langle \psi, \phi \rangle, \quad \forall\, |\phi\rangle \in \mathcal{H}.$$

Remark 1.1. The mapping $|\psi\rangle \mapsto \langle\psi|$ is an antilinear isometric isomorphism $J : \mathcal{H} \to \mathcal{H}^*$, where $\mathcal{H}^*$ is the dual space.

1.1.1 Adjoint, Hermitian, and Unitary Operators

Definition 1.4 (Adjoint)**.** The **adjoint** $L^\dagger$ satisfies $\langle\phi|\, L\,|\psi\rangle = \langle L^\dagger\phi|\,\psi$ for all states.

Definition 1.5 (Hermitian Operator)**.** An operator H is **Hermitian** if $H = H^\dagger$. Its eigenvalues are real, and eigenvectors corresponding to distinct eigenvalues are orthogonal.

Definition 1.6 (Unitary Operator)**.** An operator U is **unitary** if $U^\dagger U = I$. Unitary operators preserve norms and inner products.

1.2 Eigenvalues and Eigenvectors

Definition 1.7 (Eigenvalue Equation)**.** For L a linear operator, an eigenvector $|v\rangle$ and eigenvalue λ satisfy

$$L\,|v\rangle = \lambda\,|v\rangle\,.$$

Spectral decomposition: any Hermitian H can be written as $H = \sum_i h_i\,|h_i\rangle\,\langle h_i|$.

1.3 Quantum States and Observables

A quantum state is represented by a normalized vector $|\psi\rangle$. Observables correspond to Hermitian operators; measurement yields an eigenvalue with probability $|\,\langle h_i|\psi|h_i|\psi\rangle\,|^2$.

1.4 Quantum Gates as Unitary Transformations

Elementary gates act on qubits (2-dimensional Hilbert spaces):

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \qquad Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \qquad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix},$$

$$H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}.$$

These are unitary and represent rotation and superposition operations.

Definition 1.8 (Standard and Hadamard Quantum States)**.** Let $\mathcal{H} = \mathbb{C}^2$ be the single-qubit Hilbert space with the canonical orthonormal basis

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

We then define the **Hadamard basis** states by applying the Hadamard operator

$$H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

to the computational basis as follows:

$$|+\rangle \;=\; H\,|0\rangle \;=\; \frac{1}{\sqrt{2}}\big(|0\rangle + |1\rangle\big) = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ 1 \end{pmatrix},$$

$$|-\rangle \;=\; H\,|1\rangle \;=\; \frac{1}{\sqrt{2}}\big(|0\rangle - |1\rangle\big) = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ -1 \end{pmatrix}.$$

These four vectors satisfy the following orthonormality relations:

$$\langle 0 \mid 0\rangle = \langle 1 \mid 1\rangle = \langle + \mid +\rangle = \langle - \mid -\rangle = 1,$$

$$\langle 0 \mid 1 \rangle = \langle + \mid - \rangle = 0,$$

and in fact

$$\langle 0 \mid + \rangle = \langle 0 \mid - \rangle = \langle 1 \mid + \rangle = \langle 1 \mid - \rangle = \tfrac{1}{\sqrt{2}}(\pm 1),$$

so that each of $|0\rangle, |1\rangle, |+\rangle, |-\rangle$ has unit norm:

$$\| \left|\psi\right\rangle \| = \sqrt{\langle \psi | \psi \rangle} = 1 \quad \text{for } \left|\psi\right\rangle \in \{|0\rangle, |1\rangle, |+\rangle, |-\rangle\}.$$

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \quad |+\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \quad |-\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ -1 \end{pmatrix}.$$

Norms:

$$\langle 0|0\rangle = \begin{pmatrix} 1 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = 1, \quad \langle 1|1\rangle = (0\ 1) \begin{pmatrix} 0 \\ 1 \end{pmatrix} = 1,$$

$$\langle +|+\rangle = \frac{1}{2}(1\ 1) \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \frac{1}{2}(1+1) = 1, \quad \langle -|- \rangle = \frac{1}{2}(1\ \ -1) \begin{pmatrix} 1 \\ -1 \end{pmatrix} = \frac{1}{2}(1+1) = 1.$$

Computational–Hadamard overlaps:

$$\langle 0|+\rangle = (1\ 0)\frac{1}{\sqrt{2}}\begin{pmatrix}1\\1\end{pmatrix} = \frac{1}{\sqrt{2}}(1\cdot 1 + 0\cdot 1) = \frac{1}{\sqrt{2}},$$

$$\langle 0|-\rangle = (1\ 0)\frac{1}{\sqrt{2}}\begin{pmatrix}1\\-1\end{pmatrix} = \frac{1}{\sqrt{2}}(1\cdot 1 + 0\cdot(-1)) = \frac{1}{\sqrt{2}},$$

$$\langle 1|+\rangle = (0\ 1)\frac{1}{\sqrt{2}}\begin{pmatrix}1\\1\end{pmatrix} = \frac{1}{\sqrt{2}}(0\cdot 1 + 1\cdot 1) = \frac{1}{\sqrt{2}},$$

$$\langle 1|-\rangle = (0\ 1)\frac{1}{\sqrt{2}}\begin{pmatrix}1\\-1\end{pmatrix} = \frac{1}{\sqrt{2}}(0\cdot 1 + 1\cdot(-1)) = -\frac{1}{\sqrt{2}}.$$

Computational–computational and Hadamard–Hadamard orthogonality:

$$\langle 0|1\rangle = (1\ 0)\begin{pmatrix}0\\1\end{pmatrix} = 0, \quad \langle +|-\rangle = \frac{1}{2}(1\ 1)\begin{pmatrix}1\\-1\end{pmatrix} = \frac{1}{2}(1-1) = 0.$$

All other overlaps follow by conjugation or symmetry. Thus the set $\{|0\rangle, |1\rangle, |+\rangle, |-\rangle\}$ is orthonormal and each has unit norm.

1.5 Eigenvalue Problems

Definition 1.9 (Eigenvalues and Eigenvectors)**.** For L a linear operator, a scalar $\lambda \in \mathbb{C}$ and nonzero ket $|v\rangle$ satisfying

$$L|v\rangle = \lambda|v\rangle$$

are called an **eigenvalue** and its corresponding **eigenvector**.

Spectral theorems guarantee diagonalizability of Hermitian and normal operators.

1.6 Tensor Products

For composite quantum systems, state spaces combine via the tensor product.

Definition 1.10 (Tensor Product of Spaces)**.** Given Hilbert spaces $\mathcal{H}_A$ and $\mathcal{H}_B$, their tensor product $\mathcal{H}_A \otimes \mathcal{H}_B$ is the completion of the span of simple tensors $\left|\psi\right\rangle_A \otimes \left|\phi\right\rangle_B$ under the inner product

$$\left\langle \psi_A \otimes \phi_B|\psi'_A \otimes \phi'_B\middle|\psi_A \otimes \phi_B|\psi'_A \otimes \phi'_B\right\rangle = \left\langle \psi_A|\psi'_A\middle|\psi_A|\psi'_A\right\rangle \left\langle \phi_B|\phi'_B\middle|\phi_B|\phi'_B\right\rangle.$$

Definition 1.11 (Tensor Product of Operators)**.** For $A \in \mathcal{L}(\mathcal{H}_A)$ and $B \in \mathcal{L}(\mathcal{H}_B)$, define $A \otimes B \in \mathcal{L}(\mathcal{H}_A \otimes \mathcal{H}_B)$ by

$$(A \otimes B)(\left|\psi\right\rangle_A \otimes \left|\phi\right\rangle_B) = (A\left|\psi\right\rangle_A) \otimes (B\left|\phi\right\rangle_B).$$

1.7 Matrix Representations and Quantum Gates

In an orthonormal basis $\{|i\rangle\}$, operators and kets admit matrix and column-vector representations. Common single-qubit gates include:

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

Multi-qubit gates arise as tensor products of these.

Definition 1.12 (Action of Single-Qubit Gates on Canonical States). Let $\mathcal{H} = \operatorname{span}\{|0\rangle, |1\rangle\}$ be the single-qubit Hilbert space, and define

$$|+\rangle = \tfrac{1}{\sqrt{2}}(|0\rangle + |1\rangle), \qquad |-\rangle = \tfrac{1}{\sqrt{2}}(|0\rangle - |1\rangle).$$

The Pauli and Hadamard gates act on these four states as follows:

$$\begin{aligned} X|0\rangle &= |1\rangle, & X|1\rangle &= |0\rangle, \\ X|+\rangle &= |+\rangle, & X|-\rangle &= -|-\rangle, \\ Y|0\rangle &= i|1\rangle, & Y|1\rangle &= -i|0\rangle, \\ Y|+\rangle &= i|-\rangle, & Y|-\rangle &= -i|+\rangle, \\ Z|0\rangle &= |0\rangle, & Z|1\rangle &= -|1\rangle, \\ Z|+\rangle &= |-\rangle, & Z|-\rangle &= |+\rangle, \\ H|0\rangle &= |+\rangle, & H|1\rangle &= |-\rangle, \\ H|+\rangle &= |0\rangle, & H|-\rangle &= |1\rangle. \end{aligned}$$

In matrix form (in the $\{|0\rangle, |1\rangle\}$ basis),

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

Definition 1.13 (Hermitian Operator). An operator H on a Hilbert space $\mathcal{H}$ is called **Hermitian** (or self–adjoint) if

$$H^\dagger = H.$$

Equivalently, for all $|\psi\rangle, |\phi\rangle \in \mathcal{H}$,

$$\langle\psi| H |\phi\rangle = \overline{\langle\phi| H |\psi\rangle}.$$

Definition 1.14 (Unitary Operator). An operator U on $\mathcal{H}$ is called **unitary** if

$$U^\dagger U = U U^\dagger = I.$$

Equivalently, $U^{-1} = U^\dagger$, and U preserves inner products:

$$\langle U\psi| U\phi = \langle\psi| \phi.$$

1.8 Conclusion

This linear algebra toolkit underpins quantum algorithm design and analysis. Mastery of these concepts is essential for advanced study in quantum computation and information.

Chapter 2

Basic Quantum Algorithms

This chapter presents three foundational quantum algorithms:

- Deutsch's algorithm,
- the Deutsch–Jozsa algorithm, and
- the Bernstein–Vazirani algorithm.

Each illustrates how quantum interference and phase-kickback enable exponential or polynomial speedups over classical counterparts.

2.1 Deutsch's Algorithm

2.1.1 Problem Statement

Definition 2.1 (Boolean Oracle Problem)**.** Let $f : \{0,1\} \to \{0,1\}$ be a black-box Boolean function. We are promised that f is either **constant** (i.e. $f(0) = f(1)$) or **balanced** (i.e. $f(0) \neq f(1)$). The goal is to decide which case holds using as few queries to f as possible.

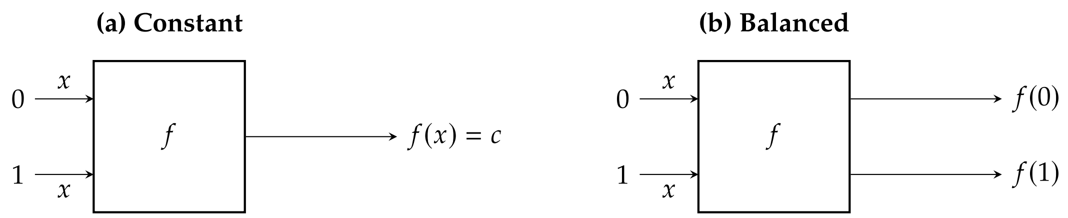


Figure 2.1: Oracle for the constant vs. balanced promise problem.

Classically, one must query both $f(0)$ and $f(1)$ to distinguish the two cases, yielding a lower bound of two queries. Deutsch's quantum algorithm achieves a one-query solution by exploiting superposition and interference.

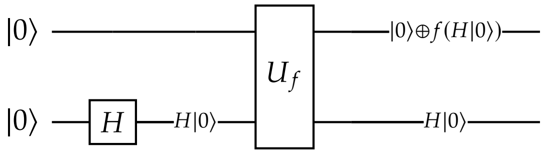
2.1.2 Quantum Oracle Model

Definition 2.2 (Oracle Unitary on Two Qubits)**.** Given a Boolean function $f : \{0,1\} \longrightarrow \{0,1\}$, the associated **oracle unitary** is the linear operator

$$U_f \quad : \quad \begin{array}{ccc} \mathbb{C}^4 & \longrightarrow & \mathbb{C}^4 \\ \left|x, y\right\rangle & \longmapsto & \left|x, y \oplus f(x)\right\rangle \end{array},$$

where $\mathbb{C}^4 = \mathbb{C}^2 \otimes \mathbb{C}^2$ is the two-qubit Hilbert space and $x, y \in \{0,1\}$. This preserves unitarity and acts trivially on superpositions.

Example 2.1. Given $U_f : \mathbb{C}^4 \to \mathbb{C}^4 : \left|x, y\right\rangle \mapsto \left|x, y \oplus f(x)\right\rangle$, consider :



Then

$$\begin{aligned} U_f(H\left|0\right\rangle, \left|0\right\rangle) = U_f(\left|+\right\rangle \otimes \left|0\right\rangle) = U_f\left(\tfrac{\left|0\right\rangle+\left|1\right\rangle}{\sqrt{2}} \otimes \left|0\right\rangle\right) &= \frac{1}{\sqrt{2}}\Big(U_f\left|0,0\right\rangle + U_f\left|1,0\right\rangle\Big) \quad \text{by linearity} \\ &= \frac{1}{\sqrt{2}}\Big(\left|0, 0 \oplus f(0)\right\rangle + \left|1, 0 \oplus f(1)\right\rangle\Big) \\ &= \frac{\left|0, f(0)\right\rangle + \left|1, f(1)\right\rangle}{\sqrt{2}}. \end{aligned}$$

- If f is constant ($f(0) = f(1) = 0$),

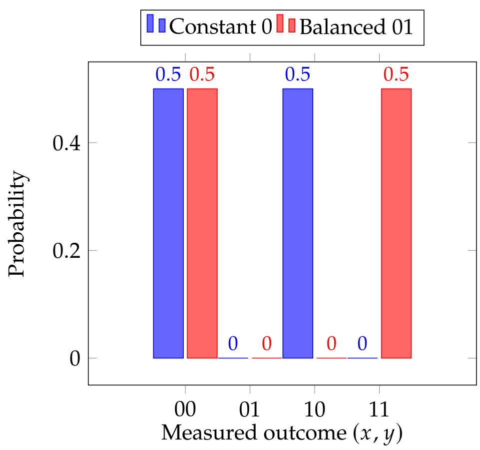
$$\left|\psi_{\text{out}}\right\rangle = U_f(\left|+\right\rangle, \left|0\right\rangle) = \frac{\left|0,0\right\rangle + \left|1,0\right\rangle}{\sqrt{2}} = \frac{1}{\sqrt{2}}\left|00\right\rangle \; + \; \frac{1}{\sqrt{2}}\left|10\right\rangle \; + \; 0 \cdot \left|01\right\rangle \; + \; 0 \cdot \left|11\right\rangle.$$

- If f is balanced ($f(0) = 0, f(1) = 1$),

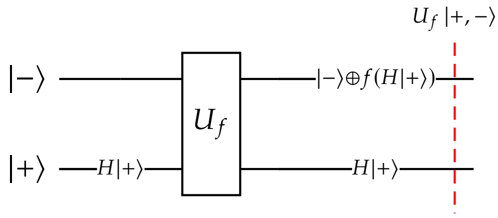
$$\left|\psi_{\text{out}}\right\rangle = U_f(\left|+\right\rangle, \left|0\right\rangle) = \frac{\left|0,0\right\rangle + \left|1,1\right\rangle}{\sqrt{2}} = \frac{1}{\sqrt{2}}\left|00\right\rangle \; + \; 0 \cdot \left|10\right\rangle \; + \; 0 \cdot \left|01\right\rangle \; + \; \frac{1}{\sqrt{2}} \cdot \left|11\right\rangle,$$

which is an entangled two-qubit state.

Outcome (x, y)	$\left\|00\right\rangle$	$\left\|01\right\rangle$	$\left\|10\right\rangle$	$\left\|11\right\rangle$
Constant ($f(0) = f(1) = 0$)	1/2	0	1/2	0
Balanced ($f(0) = 0, f(1) = 1$)	1/2	0	0	1/2



Example 2.2. Given $U_f : \mathbb{C}^4 \to \mathbb{C}^4 : \left|x, y\right\rangle \mapsto \left|x, y \oplus f(x)\right\rangle$, consider:



Then

$$\begin{aligned}
\left|+,-\right\rangle &= H\left|0\right\rangle \otimes H\left|1\right\rangle = H\left|0\right\rangle \otimes H(X\left|0\right\rangle), \\
\left|+,-\right\rangle &= \left(\frac{\left|0\right\rangle + \left|1\right\rangle}{\sqrt{2}}\right) \otimes \left(\frac{\left|0\right\rangle - \left|1\right\rangle}{\sqrt{2}}\right) = \frac{1}{2}\left(\left|00\right\rangle - \left|01\right\rangle + \left|10\right\rangle - \left|11\right\rangle\right),
\end{aligned}$$

and so

$$U_f\left|+,-\right\rangle = \frac{1}{2}\Big(U_f\left|00\right\rangle - U_f\left|01\right\rangle + U_f\left|10\right\rangle - U_f\left|11\right\rangle\Big),$$

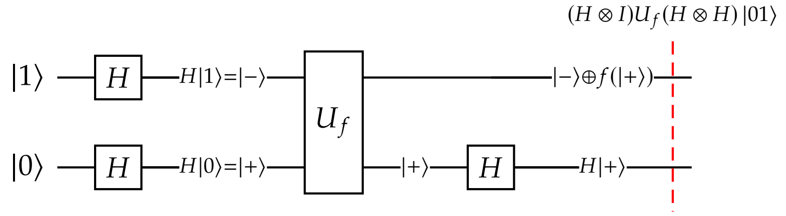
where

$$\begin{aligned}
U_f\left|00\right\rangle &= \left|0, 0 \oplus f(0)\right\rangle &&= (1 - f(0))\left|00\right\rangle + f(0)\left|01\right\rangle, \\
U_f\left|01\right\rangle &= \left|0, 1 \oplus f(0)\right\rangle &&= f(0)\left|00\right\rangle + (1 - f(0))\left|01\right\rangle, \\
U_f\left|10\right\rangle &= \left|1, 0 \oplus f(1)\right\rangle &&= (1 - f(1))\left|10\right\rangle + f(1)\left|11\right\rangle, \\
U_f\left|11\right\rangle &= \left|1, 1 \oplus f(1)\right\rangle &&= f(1)\left|10\right\rangle + (1 - f(1))\left|11\right\rangle.
\end{aligned}$$

Thus, we have

$$\begin{aligned}
U_f\left|+,-\right\rangle &= \frac{1}{2}\Big(U_f\left|00\right\rangle - U_f\left|01\right\rangle + U_f\left|10\right\rangle - U_f\left|11\right\rangle\Big) \\
&= \frac{1}{2}\big[(1 - 2f(0))\left|00\right\rangle + (f(0) - (1 - f(0)))\left|01\right\rangle + (1 - 2f(1))\left|10\right\rangle + (f(1) - (1 - f(1)))\left|11\right\rangle\big] \\
&= \frac{1}{2}\big[(1 - 2f(0))\left|00\right\rangle + (-1 + 2f(0))\left|01\right\rangle + (1 - 2f(1))\left|10\right\rangle + (-1 + 2f(1))\left|11\right\rangle\big] \\
&= \frac{1}{2}\big[(1 - 2f(0))(\left|00\right\rangle - \left|01\right\rangle) + (1 - 2f(1))(\left|10\right\rangle - \left|11\right\rangle)\big] \\
&= \frac{1}{2}\big[(1 - 2f(0))\left|0\right\rangle \otimes (\left|0\right\rangle - \left|1\right\rangle) + (1 - 2f(1))\left|1\right\rangle \otimes (\left|0\right\rangle - \left|1\right\rangle)\big] \\
&= \frac{1}{\sqrt{2}}\big[(1 - 2f(0))\left|0\right\rangle + (1 - 2f(1))\left|1\right\rangle\big] \otimes \frac{(\left|0\right\rangle - \left|1\right\rangle)}{\sqrt{2}} \\
&= \frac{1}{\sqrt{2}}\big[(1 - 2f(0))\left|0\right\rangle + (1 - 2f(1))\left|1\right\rangle\big] \otimes \left|-\right\rangle.
\end{aligned}$$

Example 2.3. Given $U_f : \mathbb{C}^4 \to \mathbb{C}^4 : \left|x, y\right\rangle \mapsto \left|x, y \oplus f(x)\right\rangle$, consider $U_f(H\left|0\right\rangle, \left|0\right\rangle)$:



Then

$$
\begin{aligned}
(H \otimes I)U_f\left|+,-\right\rangle &= \frac{H \otimes I}{\sqrt{2}}\left[(1-2f(0))\left|0\right\rangle + (1-2f(1))\left|1\right\rangle\right] \otimes \left|-\right\rangle \\
&= \frac{1}{\sqrt{2}}\left[(1-2f(0))H\left|0\right\rangle + (1-2f(1))H\left|1\right\rangle\right] \otimes I\left|-\right\rangle \\
&= \frac{1}{\sqrt{2}}\left[(1-2f(0))\left|+\right\rangle + (1-2f(1))\left|-\right\rangle\right] \otimes \left|-\right\rangle \\
&= \frac{1}{\sqrt{2}}\left[(1-2f(0))\left(\frac{\left|0\right\rangle + \left|1\right\rangle}{\sqrt{2}}\right) + (1-2f(1))\left(\frac{\left|0\right\rangle - \left|1\right\rangle}{\sqrt{2}}\right)\right] \otimes \left|-\right\rangle \\
&= \frac{1}{2}\left[(1-2f(0))(\left|0\right\rangle + \left|1\right\rangle) + (1-2f(1))(\left|0\right\rangle - \left|1\right\rangle)\right] \otimes \left|-\right\rangle \\
&= \frac{1}{2}\left[(1-2f(0)+1-2f(1))\left|0\right\rangle + (1-2f(0)+1-2f(1))\left|1\right\rangle\right] \otimes \left|-\right\rangle \\
&= \left[(1-f(0)-f(1))\left|0\right\rangle + (1-f(0)-f(1))\left|1\right\rangle\right] \otimes \left|-\right\rangle,
\end{aligned}
$$

and so

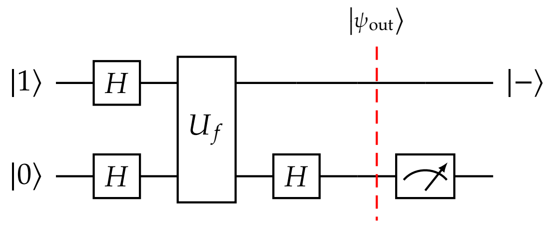
$$
\begin{aligned}
(H \otimes I)U_f(H \otimes H)\left|0,1\right\rangle &= (H \otimes I)U_f\left|+,-\right\rangle \\
&= \left[(1-f(0)-f(1))\left|0\right\rangle + (1-f(0)-f(1))\left|1\right\rangle\right] \otimes \left|-\right\rangle.
\end{aligned}
$$

Let

$$\left|\psi_{\text{in}}\right\rangle = (H \otimes H)\left|0,1\right\rangle = \left|+\right\rangle \otimes \left|-\right\rangle,$$

and let

$$\left|\psi'\right\rangle = U_f\left|\psi_{\text{in}}\right\rangle = \frac{1}{2}\Big((-1)^{f(0)}\left|0,0\right\rangle - (-1)^{f(0)}\left|0,1\right\rangle + (-1)^{f(1)}\left|1,0\right\rangle - (-1)^{f(1)}\left|1,1\right\rangle\Big).$$



Then applying H to the first qubit yields

$$\left|\psi_{\text{out}}\right\rangle = (H \otimes I)\left|\psi'\right\rangle = \begin{cases} (-1)^{f(0)}\left|0\right\rangle \otimes \left|-\right\rangle, & \text{if } f(0) = f(1) \text{ (constant)}, \\ (-1)^{f(0)}\left|1\right\rangle \otimes \left|-\right\rangle, & \text{if } f(0) \neq f(1) \text{ (balanced)}. \end{cases}$$

In particular, up to the irrelevant global phase $(-1)^{f(0)}$, the final state is $\left|0\right\rangle \otimes \left|-\right\rangle$ exactly when f is constant, and $\left|1\right\rangle \otimes \left|-\right\rangle$ exactly when f is balanced.

2.1.3 Algorithm

Note. Note that

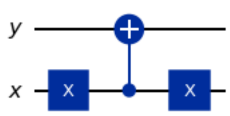
1. Initialize two qubits in $|0\rangle \otimes |1\rangle$.
2. Apply Hadamard gates to both: $H^{\otimes 2}|0, 1\rangle = |+, -\rangle$.
3. Query the oracle: $|\psi_1\rangle = U_f |+, -\rangle$.
4. Apply $H \otimes I$, yielding $|\psi_{out}\rangle = (H \otimes I)|\psi_1\rangle$.
5. Measure the first qubit in the computational basis.

Implementation of Deutsch Algorithm

```
@:~$ !pip install qiskit qiskit-ibm-runtime pylatexenc qiskit_aer
```

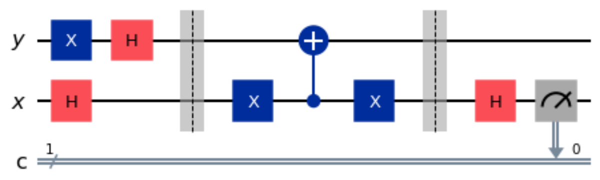
```python
from qiskit import QuantumCircuit, QuantumRegister, ClassicalRegister # Qiskit imports
from numpy import random # random number generator

def deutsch_oracle_circuit(): # Deutsch oracle builder
    qy = QuantumRegister(1, 'y') # output qubit
    qx = QuantumRegister(1, 'x') # input qubit
    qc = QuantumCircuit(qy, qx) # init 2-qubit circuit

    random.seed() # seed RNG
    f = random.randint(4) # choose f in {0,1,2,3}
    print('random number : ', f) # debug print

    if f == 0:
        pass # f=0: do nothing
    elif f == 1:
        qc.x(qy) # f=1: flip y
    elif f == 2:
        qc.cx(qx, qy) # f=2: CNOT x->y
    else:
        # f=3: X-CNOT-X on x
        qc.x(qx)
        qc.cx(qx, qy)
        qc.x(qx)

    qc.name = 'Deutsch' # label circuit
    return qc # return oracle

circuit = deutsch_oracle_circuit() # build oracle
circuit.draw(output='mpl') # draw with matplotlib
```

```
random number : 3
```



```
qx = QuantumRegister(1, 'x') # input qubit
qy = QuantumRegister(1, 'y') # output qubit
c = ClassicalRegister(1, 'c') # classical bit

circuit = QuantumCircuit(qy, qx, c)
circuit.h(qx) # superpose input
circuit.x(qy) # prepare output in |1>
circuit.h(qy) # superpose output
circuit.barrier() # separate stages
oracle = deutsch_oracle_circuit()
circuit.compose(oracle, [qy[0], qx[0]], inplace=True) # apply oracle
circuit.barrier() # separate stages
circuit.h(qx) # interference on input

circuit.measure(qx, c) # measure input
circuit.draw(output='mpl') # visualize circuit
```

```
1 random number : 3
```




```
# simulate circuit with AerSimulator and print counts
from qiskit.transpiler.preset_passmanagers import generate_preset_pass_manager
from qiskit_ibm_runtime import SamplerV2 as Sampler
from qiskit_aer import AerSimulator

aer_sim = AerSimulator() # create simulator
pm = generate_preset_pass_manager(backend=aer_sim, optimization_level=1)

isa_circuit = pm.run(circuit) # transpile circuit
sampler = Sampler(mode=aer_sim) # create sampler
job = sampler.run([isa_circuit], shots=1) # run circuit once
result = job.result() # get results
count = result[0].data.c.get_counts() # extract counts
print(count) # print counts
```

```
1 {'1': 1}
```

```
if ('0' in count) :
    answer = 'constant'
else :
    answer = 'balanced'

print (f'f(x) is a {answer} function.')
```

```
1 f(x) is a balanced function.
```

2.2 Phase Kickback

In many quantum algorithms, such as Deutsch, Deutsch–Jozsa, and Bernstein–Vazirani, the **phase kickback** effect allows one to encode information about a Boolean function f into relative phases of a register. We present here a rigorous derivation and formal statement of phase kickback.

2.2.1 Oracle Unitary and Hadamard Eigenstate

Let $f : \{0,1\}^n \to \{0,1\}$ be a Boolean function, and define the **oracle unitary**

$$U_f : \mathbb{C}^{2^n} \otimes \mathbb{C}^2 \to \mathbb{C}^{2^n} \otimes \mathbb{C}^2, \quad U_f\big(|x\rangle \otimes |y\rangle\big) = |x\rangle \otimes |y \oplus f(x)\rangle ,$$

where $x \in \{0,1\}^n$, $y \in \{0,1\}$. As shown in Section **??**, U_f is unitary and acts on the second qubit by a conditional bit-flip.

Define the Hadamard eigenstates on the second qubit:

$$|+\rangle = H|0\rangle = \tfrac{1}{\sqrt{2}}(|0\rangle + |1\rangle), \qquad |-\rangle = H|1\rangle = \tfrac{1}{\sqrt{2}}(|0\rangle - |1\rangle).$$

Lemma 2.1 (Phase Kickback)**.** *For any $x \in \{0,1\}^n$, the oracle unitary satisfies*

$$U_f\big(|x\rangle \otimes |-\rangle\big) = |x\rangle \otimes X^{f(x)}|-\rangle = (-1)^{f(x)}|x\rangle \otimes |-\rangle .$$

Proof. Since $X|-\rangle = -|-\rangle$ and $X^0 = I$, we have

$$X^{f(x)}|-\rangle = \begin{cases} |-\rangle, & f(x) = 0, \\ -|-\rangle, & f(x) = 1, \end{cases} = (-1)^{f(x)}|-\rangle .$$

Hence

$$U_f(|x\rangle \otimes |-\rangle) = |x\rangle \otimes X^{f(x)}|-\rangle = (-1)^{f(x)}|x\rangle \otimes |-\rangle .$$

□

2.2.2 Multi-Qubit Extension

By linearity, an arbitrary first-register state $|\psi\rangle = \sum_x a_x |x\rangle$ yields

$$U_f\big(|\psi\rangle \otimes |-\rangle\big) = \sum_x a_x(-1)^{f(x)}|x\rangle \otimes |-\rangle = \big(V_f|\psi\rangle\big) \otimes |-\rangle ,$$

where the **phase oracle** is

$$V_f = \sum_x (-1)^{f(x)}|x\rangle\langle x| .$$

2.2.3 Applications and Remarks

- In the **Deutsch** algorithm ($n = 1$), one starts with $(H \otimes I)U_f(H \otimes H)|0,1\rangle$, and phase kickback yields $(H \otimes I)U_f(|+\rangle \otimes |-\rangle) = \pm|0\rangle \otimes |-\rangle$ or $\pm|1\rangle \otimes |-\rangle$, distinguishing constant versus balanced cases with one final Hadamard and measurement.

- In **Deutsch–Jozsa** and **Bernstein–Vazirani**, phase kickback on an n-qubit superposition imprints the global phase pattern $(-1)^{f(x)}$, subsequently decoded by an n-fold Hadamard transform.

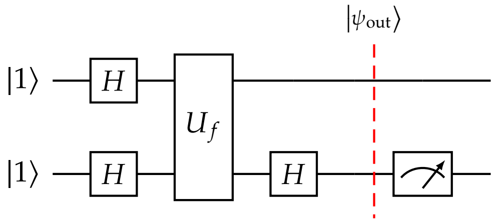
- The phase oracle V_f acts without extra ancilla, showing that any y-qubit initialized to $|-\rangle$ serves as a pure phase marker.

This completes the formal lecture notes on phase kickback, a key primitive in quantum algorithmic speed-ups.

2.2.4 Exercises

Consider the two-qubit unitary evolution

$$\left|\psi_{\text{in}}\right\rangle = (H \otimes H)\left|1,1\right\rangle, \quad \left|\psi'\right\rangle = U_f \left|\psi_{\text{in}}\right\rangle, \quad \left|\psi_{\text{out}}\right\rangle = (H \otimes I)\left|\psi'\right\rangle.$$



(a) Show that

$$\left|\psi_{\text{out}}\right\rangle = \frac{1}{2\sqrt{2}}\Big[(-1)^{f(0)}\left|00\right\rangle - (-1)^{f(0)}\left|01\right\rangle + (-1)^{f(1)}\left|10\right\rangle - (-1)^{f(1)}\left|11\right\rangle\Big],$$

and hence write $\left|\psi_{\text{out}}\right\rangle$ as a linear combination of the basis $\{\left|00\right\rangle, \left|01\right\rangle, \left|10\right\rangle, \left|11\right\rangle\}$.

(b) Suppose $f(0) = f(1) = 1$. If the first qubit (q_1) is measured in the computational basis, compute $\Pr\big[q_1 = 1\big]$ and $\Pr\big[q_1 = 0\big]$.

Sol. (a) First,

$$\left|\psi_{\text{in}}\right\rangle = (H \otimes H)\left|1,1\right\rangle = \left|-\right\rangle \otimes \left|-\right\rangle = \frac{1}{2}\sum_{x=0}^{1}\left|x\right\rangle \otimes (\left|0\right\rangle - \left|1\right\rangle).$$

By phase kickback,

$$\left|\psi'\right\rangle = U_f(\left|-\right\rangle \otimes \left|-\right\rangle) = \frac{1}{2}\sum_{x=0}^{1}(-1)^{f(x)}\left|x\right\rangle \otimes (\left|0\right\rangle - \left|1\right\rangle).$$

Applying H on the first qubit gives

$$\left|\psi_{\text{out}}\right\rangle = (H \otimes I)\left|\psi'\right\rangle = \frac{1}{2}\sum_{x=0}^{1}(-1)^{f(x)}(H\left|x\right\rangle) \otimes (\left|0\right\rangle - \left|1\right\rangle) = \frac{1}{2\sqrt{2}}\sum_{a,x=0}^{1}(-1)^{ax+f(x)}\left|a\right\rangle \otimes (\left|0\right\rangle - \left|1\right\rangle),$$

which is equivalently

$$\boxed{\left|\psi_{\text{out}}\right\rangle = \frac{1}{2\sqrt{2}}\big[(-1)^{f(0)}\left|00\right\rangle - (-1)^{f(0)}\left|01\right\rangle + (-1)^{f(1)}\left|10\right\rangle - (-1)^{f(1)}\left|11\right\rangle\big].}$$

(b) If $f(0) = f(1) = 1$, then $(-1)^{f(x)} = -1$ for $x = 0, 1$. Substituting,

$$\left|\psi_{\text{out}}\right\rangle = \frac{-1}{2\sqrt{2}}\big[\left|00\right\rangle - \left|01\right\rangle + \left|10\right\rangle - \left|11\right\rangle\big] = \frac{-1}{\sqrt{2}}(\left|00\right\rangle - \left|01\right\rangle).$$

Thus the only nonzero amplitudes are on $\left|00\right\rangle$ and $\left|01\right\rangle$, each of magnitude $1/\sqrt{2}$. Consequently,

$$\Pr\big[q_1 = 0\big] = \left(-\frac{1}{\sqrt{2}}\right)^2 + \left(-\frac{1}{\sqrt{2}}\right)^2 = \frac{1}{2} + \frac{1}{2} = 1, \quad \Pr\big[q_1 = 1\big] = 0.$$

This completes the solution. □

2.3 Deutsch–Jozsa Algorithm

2.3.1 Problem Statement

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

- **constant**: $f(x)$ same for all x, or
- **balanced**: $f(x) = 0$ for exactly 2^{n-1} inputs,

Classically requires $2^{n-1} + 1$ evaluations in worst case; quantum requires one.

2.3.2 Algorithm Steps

1. Prepare $n + 1$ qubits in $|0\rangle^{\otimes n} \otimes |1\rangle$.
2. Apply $H^{\otimes(n+1)}$, yielding $\left|\psi_1\right\rangle = \frac{1}{2^{n/2}} \sum_x |x\rangle \otimes |-\rangle$.
3. Oracle: $\left|\psi_2\right\rangle = U_f \left|\psi_1\right\rangle = \frac{1}{2^{n/2}} \sum_x (-1)^{f(x)} |x\rangle \otimes |-\rangle$.
4. Apply $H^{\otimes n} \otimes I$, obtaining

$$\left|\psi_{out}\right\rangle = \frac{1}{2^n} \sum_{y,x} (-1)^{f(x)+x\cdot y} \left|y\right\rangle \otimes |-\rangle .$$

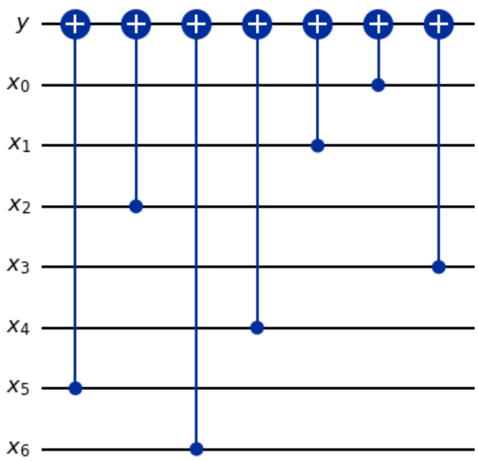
5. Measure the first n qubits.

2.3.3 Analysis

The amplitude on $|0^n\rangle$ is $\dfrac{1}{2^n} \displaystyle\sum_x (-1)^{f(x)}$, which equals ± 1 if f is constant and 0 if balanced. A single measurement thus solves the problem with certainty.

```python
from qiskit import QuantumCircuit, QuantumRegister, ClassicalRegister # imports
import numpy as np # numpy

def dj_oracle_random(n): # random n-qubit DJ oracle
    qy = QuantumRegister(1, 'y') # y qubit
    qx = QuantumRegister(n, 'x') # x qubits
    qc = QuantumCircuit(qy, qx) # init circuit

    np.random.seed() # seed RNG
    condition = np.random.choice(['constant', 'balanced']) # oracle type
    print(condition) # debug

    if condition == 'constant': # constant case
        x = np.random.randint(2) # output bit
        print(x) # debug
        # if x==1 f(x)=1, if x==0 f(x)=0
        if x == 1:
            qc.x(qy) # flip y
    else: # balanced case
        k = np.random.randint(1, n+1) # number of flips
        a = np.random.permutation(n) # permute indices
        a = a[:k] # take first k indices
        print(a) # debug
        for idx in a:
            qc.cx(qx[idx], qy[0]) # CNOT x->y

    qc.name = 'D-Jozsa' # set name
    return qc # return oracle

circuit = dj_oracle_random(7) # build oracle
circuit.draw(output='mpl') # draw circuit
```

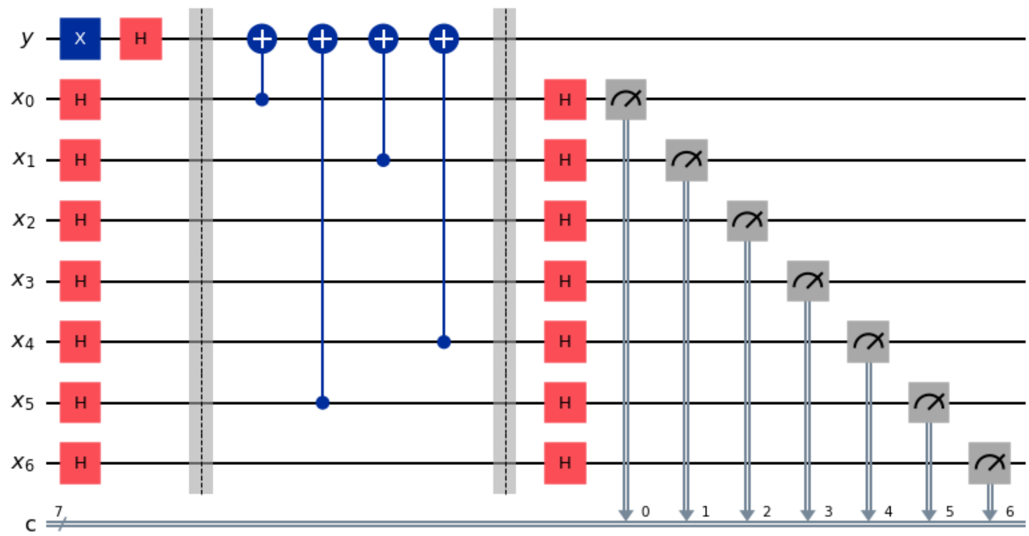
```
balanced
[5 2 6 4 1 0 3]
```




```python
n = 7
qx = QuantumRegister(n, 'x') # input qubits
qy = QuantumRegister(1, 'y') # ancilla qubit
c = ClassicalRegister(n, 'c') # classical bits
```

```
circuit = QuantumCircuit(qy, qx, c) # init circuit
for qubit in qx:
circuit.h(qubit) # H on all x qubits
circuit.x(qy) # X on y
circuit.h(qy) # H on y
circuit.barrier() # barrier
oracle_circuit = dj_oracle_random(n) # build oracle
circuit.compose(oracle_circuit, qubits=[*qy, *qx], inplace=True) # apply oracle
circuit.barrier() # barrier
for qubit in qx:
    circuit.h(qubit) # H on all x qubits
for i in range(n):
    circuit.measure(qx[i], c[i]) # measure x into c
circuit.draw(output='mpl') # draw circuit
```

```
balanced [0 5 1 4]
```




```
from qiskit.transpiler.preset_passmanagers import generate_preset_pass_manager
from qiskit_ibm_runtime import SamplerV2 as Sampler
from qiskit_aer import AerSimulator
aer_sim = AerSimulator()
pm = generate_preset_pass_manager(backend=aer_sim, optimization_level=1)
isa_circuit = pm.run(circuit)
sampler = Sampler(mode=aer_sim)
job = sampler.run([isa_circuit], shots=1)
result = job.result()
count = result[0].data.c.get_counts()
print (count)
```

```
{'0110011': 1}
```

```
n_zeros = n*'0'
if (n_zeros in count) : answer = 'constant'
else : answer = 'balanced'
print(f'f(x) is a {answer} function.')
```

```
f(x) is a balanced function.
```

2.4 Bernstein–Vazirani Algorithm

2.4.1 Problem Statement

Given oracle access to $f_s(x) = s \cdot x \pmod 2$ for unknown $s \in \{0,1\}^n$, determine s. Classically requires n queries; quantum uses one.

2.4.2 Algorithm

1. Initialize $|0\rangle^{\otimes n} \otimes |1\rangle$.
2. Apply $H^{\otimes(n+1)}$: get $\frac{1}{\sqrt{2^n}} \sum_x |x\rangle \otimes |-\rangle$.
3. Oracle: $\left|\psi_2\right\rangle = \frac{1}{\sqrt{2^n}} \sum_x (-1)^{s \cdot x} |x\rangle \otimes |-\rangle$.
4. Apply $H^{\otimes n} \otimes I$: by Fourier analysis,

$$(H^{\otimes n} \sum_x (-1)^{s \cdot x} |x\rangle) \otimes |-\rangle = |s\rangle \otimes |-\rangle \,.$$

5. Measure first n qubits to read out s.

2.4.3 Proof of Correctness

Using $H^{\otimes n} |x\rangle = \frac{1}{2^{n/2}} \sum_y (-1)^{x \cdot y} \left|y\right\rangle$, we compute

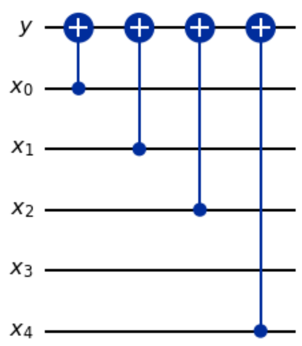
$$H^{\otimes n} \Big(\sum_x (-1)^{s \cdot x} |x\rangle \Big) = \sum_y \frac{1}{2^{n/2}} \sum_x (-1)^{s \cdot x + x \cdot y} \left|y\right\rangle = |s\rangle \,,$$

Exercises

1. Prove phase-kickback more generally: for any $f : \{0,1\}^n \to \{0,1\}$, show $U_f(|x\rangle \otimes |-\rangle) = (-1)^{f(x)} |x\rangle \otimes |-\rangle$.
2. Extend Deutsch–Jozsa to detect whether f has Hamming weight k.
3. Analyze robustness of Bernstein–Vazirani against depolarizing noise.

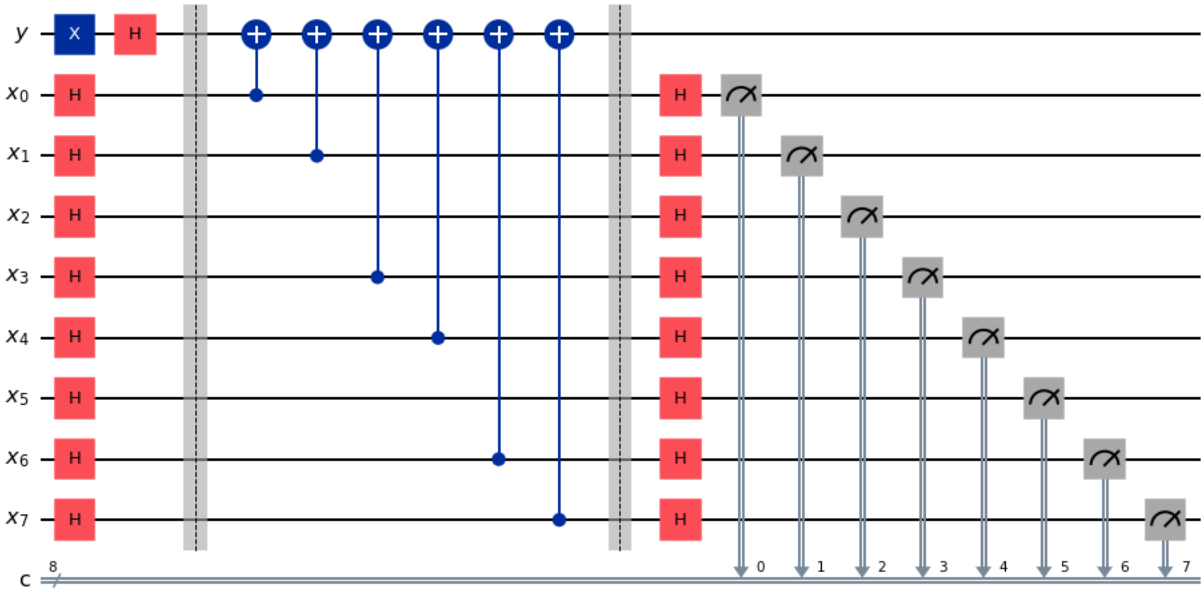
```python
from qiskit import QuantumCircuit, QuantumRegister, ClassicalRegister # Qiskit imports
import numpy as np # numpy import

# BV oracle builder
    def bv_oracle_circuit(n, reveal=False):
    qy = QuantumRegister(1, 'y') # y qubit
    qx = QuantumRegister(n, 'x') # x qubits
    qc = QuantumCircuit(qy, qx) # init circuit
    np.random.seed() # seed RNG
    s = list(np.random.randint(0, 2, size=n)) # random secret
    if reveal: print("random binary string =", s) # debug print
    for i in range(n):
        if s[i]: # secret bit check
            qc.cx(qx[i], qy[0]) # CNOT x->y
    qc.name = "BV Oracle" # label circuit
    hs = ''
    for c in reversed(s):
        hs += str(c) # build string
    return qc, hs # return circuit, secret

circuit, hs = bv_oracle_circuit(5, True) # build oracle
circuit.draw(output='mpl') # draw circuit
```

```
1  random binary string = [1, 1, 1, 0, 1].
```




```python
# build Bernstein-Vazirani circuit
n = 8
qx = QuantumRegister(n, 'x') # input qubits
qy = QuantumRegister(1, 'y') # ancilla qubit
c = ClassicalRegister(n, 'c') # classical bits
circuit = QuantumCircuit(qy, qx, c) # init circuit
circuit.h(qx) # H on x
circuit.x(qy) # X on y
circuit.h(qy) # H on y
circuit.barrier() # barrier
oracle, hs = bv_oracle_circuit(n, reveal=True) # build oracle
print(f'hidden string = {hs}') # show secret
circuit.compose(oracle, qubits=[qy[0]] + list(qx), inplace=True) # apply oracle
circuit.barrier() # barrier
circuit.h(qx) # H on x
circuit.measure(qx, c) # measure x to c
circuit.draw(output='mpl') # draw circuit
```

```
random binary string = [1, 1, 0, 1, 1, 0, 1, 1]
hidden string = 11011011
```




```
from qiskit.transpiler.preset_passmanagers import generate_preset_pass_manager
from qiskit_ibm_runtime import SamplerV2 as Sampler
from qiskit_aer import AerSimulator

aer_sim = AerSimulator()
pm = generate_preset_pass_manager(backend=aer_sim, optimization_level=1)

isa_circuit = pm.run(circuit)
sampler = Sampler(mode=aer_sim)
job = sampler.run([isa_circuit], shots=1)
result = job.result()
count = result[0].data.c.get_counts()
print (count)
```

```
{'11011011': 1}
```

```
measured = list(count.keys())[0]

print (f'The measured value {measured} ', end='')
if hs == measured :
    print(f'is equal to the hidden string {hs}')
else :
    print(f'is not equal to the hidden string {hs}')
```

```
The measured value 11011011 is equal to the hidden string 11011011
```

Appendix A

Exercises

Exercises #1

1. (Rotation Matrix in the Complex Plane). Let $\phi \in \mathbb{R}$ and consider the 2×2 matrix

$$A = \begin{pmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{pmatrix}.$$

(a) Prove that A is a unitary operator on $\mathbb{C}^2$; that is, show $A^\dagger A = I_2$, where $A^\dagger$ is the conjugate-transpose of A.

(b) Determine the full spectrum of A and exhibit for each eigenvalue a corresponding (nonzero) eigenvector.

Sol. (a) Since A has only real entries, $A^\dagger = A^T$. Then

$$\begin{aligned} A^T A &= \begin{pmatrix} \cos\phi & \sin\phi \\ -\sin\phi & \cos\phi \end{pmatrix} \begin{pmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{pmatrix} \\ &= \begin{pmatrix} \cos^2\phi + \sin^2\phi & -\cos\phi\ \sin\phi + \sin\phi\ \cos\phi \\ -\sin\phi\ \cos\phi + \cos\phi\ \sin\phi & \sin^2\phi + \cos^2\phi \end{pmatrix} \\ &= \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}. \end{aligned}$$

(b) We need to find $\lambda \in \mathbb{C}$ and nonzero $\mathbf{v} = (v_1, v_2)$ such that $A\,\mathbf{v} = \lambda\,\mathbf{v}$. Equivalently,

$$(A - \lambda I_2)\mathbf{v} = \begin{pmatrix} \cos\phi - \lambda & -\sin\phi \\ \sin\phi & \cos\phi - \lambda \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}.$$

Nontrivial solutions exist exactly when

$$\det\big(A - \lambda I\big) = (\cos\phi - \lambda)^2 + \sin^2\phi = 0.$$

Then

$$(\cos\phi - \lambda)^2 + \sin^2\phi = \lambda^2 - 2\lambda\cos\phi + (\cos^2\phi + \sin^2\phi) = \lambda^2 - 2\lambda\cos\phi + 1,$$

and so

$$\begin{aligned} \lambda^2 - 2(\cos\phi)\,\lambda + 1 = 0 \implies \lambda &= \frac{2\cos\phi \pm \sqrt{(2\cos\phi)^2 - 4\cdot 1\cdot 1}}{2} \\ \implies \lambda &= \cos\phi \pm \sqrt{\cos^2\phi - 1} \\ \implies \lambda &= \cos\phi \pm \sqrt{-\sin^2\phi} \\ \implies \lambda &= \cos\phi \pm i\sin\phi = e^{\pm i\phi}. \end{aligned}$$

Thus the two eigenvalues are

$$\lambda_1 = e^{i\phi}, \qquad \lambda_2 = e^{-i\phi}.$$

Since

$$\begin{aligned}
(A - \lambda_1 I)\mathbf{v} = 0 \implies & \begin{pmatrix} \cos\phi - (\cos\phi + i\sin\phi) & -\sin\phi \\ \sin\phi & \cos\phi - (\cos\phi + i\sin\phi) \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} = 0 \\
\implies & \begin{cases} (-\,i\sin\phi)\, v_1 \; - \; \sin\phi\, v_2 = 0, \\ \sin\phi\, v_1 \; - \; (i\sin\phi)\, v_2 = 0. \end{cases} \implies \begin{cases} -iv_1 \; - v_2 = 0, \\ v_1 \; - iv_2 = 0. \end{cases} \quad \text{if } \sin\phi \neq 0 \\
\implies & \mathbf{v} = t \begin{pmatrix} 1 \\ -i \end{pmatrix} \quad \text{with } t \neq 0,
\end{aligned}$$

a normalized eigenvector is $|v_1\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ -\,i \end{pmatrix}$. Similarly, we have $|v_2\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ i \end{pmatrix}$.

□

2. (Action of the Kronecker Product on Tensor-Product Vectors). Let

$$A \in \mathbb{C}^{n \times n}, \quad B \in \mathbb{C}^{m \times m}, \qquad \alpha \in \mathbb{C}^n, \quad \beta \in \mathbb{C}^m,$$

and form their Kronecker products $A \otimes B \in \mathbb{C}^{(nm)\times(nm)}$ and $\alpha \otimes \beta \in \mathbb{C}^{nm}$. Show that

$$(A \otimes B)\,(\alpha \otimes \beta) \;=\; (A\,\alpha) \;\otimes\; (B\,\beta).$$

Sol. 1. Let $\{e_i\}_{i=1}^n$ be the standard basis of $\mathbb{C}^n$ and $\{f_j\}_{j=1}^m$ the standard basis of $\mathbb{C}^m$. By definition of the tensor (Kronecker) product we have the basis

$$\{\, e_i \otimes f_j : 1 \le i \le n,\ 1 \le j \le m \} \quad \text{for } \mathbb{C}^n \otimes \mathbb{C}^m \cong \mathbb{C}^{nm}.$$

Write

$$\alpha = \sum_{i=1}^n \alpha_i\, e_i, \qquad \beta = \sum_{j=1}^m \beta_j\, f_j.$$

Then by bilinearity of the tensor product,

$$\alpha \otimes \beta = \sum_{i=1}^n \sum_{j=1}^m (\alpha_i\, \beta_j)\, (e_i \otimes f_j).$$

By the definition of the Kronecker-product operator,

$$(A \otimes B)\big(e_i \otimes f_j\big) = \big(A\, e_i\big) \;\otimes\; \big(B\, f_j\big),$$

and linearity then gives

$$(A \otimes B)\,(\alpha \otimes \beta) = \sum_{i=1}^n \sum_{j=1}^m \alpha_i\, \beta_j\, (A \otimes B)\big(e_i \otimes f_j\big) = \sum_{i,j} \alpha_i\, \beta_j\, \big(Ae_i \otimes Bf_j\big).$$

Observe that

$$A\alpha = A\Big(\sum_i \alpha_i e_i\Big) = \sum_i \alpha_i\, (A\, e_i), \quad B\beta = \sum_j \beta_j\, (B\, f_j).$$

Hence

$$A\alpha \;\otimes\; B\beta = \Big(\sum_i \alpha_i\, Ae_i\Big) \otimes \Big(\sum_j \beta_j\, Bf_j\Big) = \sum_{i,j} \alpha_i\, \beta_j\, \big(Ae_i \otimes Bf_j\big) = (A \otimes B)\,(\alpha \otimes \beta).$$

□

3.

4.

5. (SWAP Gate via Three CNOTs) Let

$$\mathrm{SWAP}\colon \mathbb{C}^2 \otimes \mathbb{C}^2 \to \mathbb{C}^2 \otimes \mathbb{C}^2$$

be the two-qubit operator defined on the computational basis by

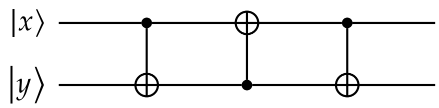
$$\mathrm{SWAP}\,\big|x, y\big\rangle \;=\; \big|y, x\big\rangle, \quad x, y \in \{0, 1\}.$$

(i) Prove that

$$\mathrm{SWAP} = \mathrm{CNOT}_{1\to 2} \,\circ\, \mathrm{CNOT}_{2\to 1} \,\circ\, \mathrm{CNOT}_{1\to 2},$$

where $\mathrm{CNOT}_{i\to j}$ denotes a CNOT gate with control qubit i and target qubit j.

(ii) Show that the following circuit indeed effects the swap of the two qubits:



Sol. Since both U and SWAP are unitary operators on the two-qubit Hilbert space, it suffices to check their action on the computational basis $\{\big|x, y\big\rangle : x, y \in \{0, 1\}\}$. Write

$$\mathrm{CNOT}_{1\to 2}\,\big|x, y\big\rangle = \big|x,\; x \oplus y\big\rangle, \quad \mathrm{CNOT}_{2\to 1}\,|a, b\rangle = |a \oplus b,\; b\rangle,$$

where $\oplus$ denotes addition modulo 2.

Step 1: Apply the first gate: $\big|x, y\big\rangle \xmapsto{\mathrm{CNOT}_{1\to 2}} \big|x,\; x \oplus y\big\rangle$.

Step 2: Apply the second gate, $\mathrm{CNOT}_{2\to 1}$, to the intermediate state:

$$\big|x,\; x \oplus y\big\rangle \xmapsto{\mathrm{CNOT}_{2\to 1}} \big|\,x \oplus (x \oplus y),\; x \oplus y\big\rangle = \big|y,\; x \oplus y\big\rangle.$$

Step 3: Finally apply $\mathrm{CNOT}_{1\to 2}$ again:

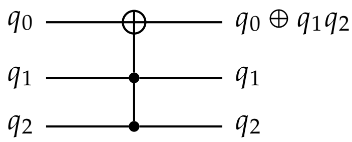
$$\big|y,\; x \oplus y\big\rangle \xmapsto{\mathrm{CNOT}_{1\to 2}} \big|y,\; y \oplus (x \oplus y)\big\rangle = \big|y,\; x\big\rangle.$$

Hence the composite action on an arbitrary basis vector is

$$U\,\big|x, y\big\rangle = \big|y, x\big\rangle,$$

which by definition is exactly SWAP $\big|x, y\big\rangle$. □

6. (Matrix Representation of the Toffoli (CCX) Gate)



The three-qubit Toffoli gate (also called the Controlled-Controlled-NOT, or CCX, gate) with qubits q_0, q_1 as controls and q_2 as target acts on the computational basis by

$$\left|q_0 q_1 q_2\right\rangle \xrightarrow{CCX_{210}} \left|(q_0 \oplus (q_1 \wedge q_2)) q_1 q_2\right\rangle, \qquad q_0, q_1, q_2 \in \{0, 1\}.$$

Using the lexicographic ordering $|000\rangle, |001\rangle, |010\rangle, \dots, |111\rangle$, represent CCX_{210} as an 8×8 unitary matrix.

Let the three-qubit Toffoli gate (also called the Controlled-Controlled-NOT, or CCX, gate) act on the computational basis $\{\left|q_0 q_1 q_2\right\rangle : q_i \in \{0, 1\}\}$ by flipping the target qubit q_2 if and only if both control qubits q_0 and q_1 are in state $|1\rangle$.

(i) Write down the action of CCX on each basis vector:

$$\mathrm{CCX}\left|q_0 q_1 q_2\right\rangle = \left|q_0\, q_1\, (q_2 \oplus (q_0 \wedge q_1))\right\rangle.$$

(ii) Using the standard ordered basis $|000\rangle, |001\rangle, |010\rangle, |011\rangle, |100\rangle, |101\rangle, |110\rangle, |111\rangle$, represent CCX as an 8×8 unitary matrix.

Sol. With the lexicographic ordering $|000\rangle, |001\rangle, |010\rangle, |011\rangle, |100\rangle, |101\rangle, |110\rangle, |111\rangle$, its matrix representation is the 8×8 unitary

$$\mathrm{CCX} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix},$$

i.e. the first six basis states are fixed and the last two are swapped. $\square$