

# Deutsch-Jozsa and Bernstein-Vazirani algorithm using single-particle discrete-time quantum walk

Ravi Sangwan<sup>1,2</sup>, Vikas Ramaswamy<sup>2</sup>, Henry Sukumar<sup>2</sup> and Gudapati Naresh Raghava<sup>2</sup>

<sup>1</sup>*Department of Physics, Indian Institute of Space Science and Technology, Thiruvananthapuram, India*

<sup>2</sup>*Quantum Technology Group, Centre for Development of Advanced Computing (C-DAC), Bangalore, India*

ravi.sc23m018@pg.iist.ac.in, vikasramaswamy@cdac.in, henrys@cdac.in, nareshraghava@cdac.in

**Abstract**—The paper introduces an efficient implementation of the Deutsch-Jozsa and Bernstein-Vazirani algorithm using the single-particle discrete-time quantum walk. We also provide a detailed optical framework with specific optical components to achieve these implementations in the photonic quantum walk scheme by simultaneously exploiting both polarization and path degrees of freedom. These implementations demonstrate improved resource efficiency while maintaining the exponential speedup characteristic of both algorithms. This work contributes to the growing field of universal quantum computing using single-particle discrete-time quantum walk.

**Index Terms**—Quantum computing, Photonic quantum walk, Deutsch-Jozsa, Bernstein-Vazirani.

## I. INTRODUCTION

Quantum computing holds the promise of revolutionizing various fields by delivering exponential improvements in the efficiency of specific computational tasks compared to classical computing [1]. Among well-established quantum algorithms, the Deutsch-Jozsa algorithm [2] and the Bernstein-Vazirani algorithm [3] are notable for their exponential speed-up over classical approaches. The Deutsch-Jozsa algorithm identifies whether a given function is constant or balanced using a single oracle query, and the Bernstein-Vazirani algorithm determines a binary string  $s$  associated with a given function in a single oracle query. Several recent theoretical advancements have extended the Deutsch-Jozsa and Bernstein-Vazirani algorithms, including distributed versions [4], generalizations beyond qubit systems [5], and supersymmetric models [6]. While these approaches improve scalability or structure, they rely on auxiliary qubits, entanglement, or complex operations, which limits their suitability for photonic platforms. In contrast, this paper presents an implementation of both algorithms using a single-particle discrete-time quantum walk, allowing the algorithm to be executed without the need for an auxiliary qubit, thereby improving resource efficiency by eliminating the additional quantum gates associated with it.

The following sections of this article provide a comprehensive exploration of the theoretical foundations and implementation details. Section II focuses on universal quantum computing using a single-particle discrete-time quantum walk along with a discussion on the use of photonic quantum walk for quantum computing. Section III delves into the theoretical background of the Deutsch-Jozsa algorithm. Section IV introduces the quantum walk-based approach to the Deutsch-Jozsa algorithm with and without using an auxiliary qubit along

with a discussion on the implementation of both approaches in a photonic system. Section V presents the theoretical background of the Bernstein-Vazirani algorithm and introduces the quantum walk-based scheme for its implementation. Section VI presents a comparative analysis of the quantum walk schemes for the Bernstein-Vazirani and Deutsch-Jozsa algorithms, examining both implementations with and without an auxiliary qubit. Finally, Section VII provides a summary and concluding remarks on the study.

## II. UNIVERSAL QUANTUM COMPUTING USING SINGLE-PARTICLE DISCRETE-TIME QUANTUM WALK

Quantum Walks (QWs) are the quantum mechanical analogues of classical random walks influenced by the principles of quantum mechanics, leading to fundamentally different behavior and properties compared to their classical counterparts. QWs can be categorized into two types: discrete-time quantum walks (DTQWs) [7] and continuous-time quantum walks (CTQWs) [8]. DTQWs describe the evolution of a quantum walker in discrete time steps. In contrast, CTQWs describe the continuous evolution of a quantum state under a Hamiltonian that dictates its dynamical evolution. In the context of DTQWs, two main categories are distinguished: coin-based models and coin-less models [9]. The coin-based models, are commonly referred to as discrete-time quantum walks (DTQWs). Coin-based quantum walks evolve under the influence of two key operators: the coin operator and the shift operator. The coin operator acts in the coin-Hilbert space  $H_c$ . Specifically, the coin operator for a 1D coin-based DTQW takes the form of a  $2 \times 2$  unitary matrix:

$$\hat{C}(p, q, r, \theta) = e^{ip} \begin{bmatrix} e^{iq} \cos(\theta) & e^{ir} \sin(\theta) \\ -e^{-ir} \sin(\theta) & e^{-iq} \cos(\theta) \end{bmatrix}, \quad (1)$$

where  $p, q, r, \theta \in \mathbb{R}$ . The shift operator acts in the position-Hilbert space  $H_s = \text{span}\{|l\rangle\}, l \in \mathbb{Z}$ , where  $\mathbb{Z}$  denotes the set of available position states. The conditional position shift operation governs the walker's movement on the basis of coin state, thereby influencing the evolution of the quantum walker in position space. The corresponding shift operators are defined as follows [10]:

$$\begin{aligned} \hat{S}_-^a &= \sum_{l \in \mathbb{Z}} \sum_b |a\rangle\langle a| \otimes |l-1\rangle\langle l| + |b \neq a\rangle\langle b| \otimes |l\rangle\langle l| \\ \hat{S}_+^b &= \sum_{l \in \mathbb{Z}} \sum_a |a \neq b\rangle\langle a| \otimes |l\rangle\langle l| + |b\rangle\langle b| \otimes |l+1\rangle\langle l|, \end{aligned} \quad (2)$$

where  $|a\rangle$  and  $|b\rangle$  represent basis states in the Hilbert space of the coin  $H_C$  ( $|a\rangle, |b\rangle \in \{|0\rangle, |1\rangle\}$ ). The set of operators  $\{\hat{I}, \hat{S}_+^0, \hat{S}_-^0, \hat{S}_+^1, \hat{S}_-^1, \hat{C}_t\}$  forms a fundamental building block for constructing 1D coin-based DTQWs. Within the DTQWs framework these set of operators can be used to effectively realize universal set of quantum gates (Hadamard ( $H$ ), Phase ( $P$ ), and CNOT). Furthermore, the coin and position states serve as a natural encoding scheme for qubits in a quantum system [10].

The scheme proposed in [11] demonstrates the experimental realization of a universal set of quantum gates (Hadamard ( $H$ ), Phase ( $P$ ), and CNOT) with high fidelity using photonic quantum walks in combination with polarization and path degrees of freedom. In this approach, qubits can be encoded using both the path and polarization degrees of freedom of a photon. The path degree of freedom corresponds to the designated path (mode) number of the photon, while the polarization degree of freedom encodes additional information, typically as horizontal ( $H$ ) or vertical ( $V$ ) polarization states. The polarization Hilbert space serves a role analogous to the coin space in DTQWs, while the path degree of freedom corresponds to the position Hilbert space. One of the qubits is encoded in the polarization degrees of freedom, while additional qubits are encoded in the path degrees of freedom, and the implementation of quantum gates is achieved through a combination of a path (mode)-dependent quantum coin operation, typically realized using a sequence of QWP-HWP-QWP (quarter-wave plate, half-wave plate, quarter-wave plate) or other waveplate configurations, along with a conditioned shift operator, which is facilitated by polarizing beam splitters (PBS) or beam splitters (BS).

### III. DEUTSCH-JOZSA ALGORITHM

The Deutsch-Jozsa algorithm signifies a substantial development in quantum computing, demonstrating an exponential speedup over classical methods for identifying whether the input function is constant or balanced [2]. The algorithm takes a function  $f : \{0, 1\}^n \rightarrow \{0, 1\}$  as input, where each  $n$ -bit input string is mapped to a binary output. The function must satisfy one of two conditions: it is either constant, giving the same output for all inputs, or balanced, giving 0 for half of the inputs and 1 for the other half. The objective is to identify whether the function is constant or balanced while minimizing the number of function evaluations. Classically, in the worst-case scenario, at least  $(2^{n-1} + 1)$  function evaluations are required. However, the Deutsch-Jozsa algorithm achieves this with a single function evaluation. For a function with an input string of  $n$ -bits, the Deutsch-Jozsa algorithm can be implemented using  $n + 1$  qubits (standard approach with auxiliary qubit) or  $n$  qubits (without auxiliary qubit). The standard approach with an auxiliary qubit follows the conventional oracle design, and the auxiliary qubit encodes the function's output in a phase. The  $n$ -qubit approach (without an auxiliary qubit) is more resource-efficient, as it eliminates the need for an extra auxiliary qubit, but requires a more intricate oracle construction that directly applies the phase transformation. The

procedure for both approaches, the standard approach (with an auxiliary qubit) and the approach without an auxiliary qubit, is outlined below:

#### 1) Standard approach (with auxiliary qubit):

- 1.1) The quantum system is prepared in an  $n + 1$  qubit state, where  $n$  qubits represent the input register and an additional auxiliary qubit serves as the function output register:

$$|\phi_0\rangle = |0\rangle^{\otimes n} \otimes |1\rangle \quad (3)$$

- 1.2) Each qubit undergoes a hadamard transformation:

$$\begin{aligned} |\phi_1\rangle &= H^{\otimes(n+1)}|\psi_0\rangle \\ &= \left( \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle \right) \otimes \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \end{aligned} \quad (4)$$

- 1.3) The function  $f$  is encoded into a quantum oracle, which applies a phase shift conditioned on the output of the function.

$$U_f|x\rangle|y\rangle = |x\rangle|y \oplus f(x)\rangle \quad (5)$$

As the auxiliary qubit is in the state  $(|0\rangle - |1\rangle)$ , the oracle introduces a phase shift of  $(-1)^{f(x)}$ :

$$U_f|\phi_1\rangle = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} (-1)^{f(x)} |x\rangle \otimes \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \quad (6)$$

- 1.4) A second Hadamard transformation is applied to the input register

$$\begin{aligned} |\phi_2\rangle &= H^{\otimes n}|\phi_1\rangle \\ &= \frac{1}{2^n} \sum_{z=0}^{2^n-1} \left( \sum_{x=0}^{2^n-1} (-1)^{x \cdot z + f(x)} \right) |z\rangle \otimes \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \end{aligned} \quad (7)$$

From (7) the probability of measuring  $z = 0$ , corresponding to  $|0\rangle^{\otimes n}$ , is

$$\left| \frac{1}{2^n} \sum_{x=0}^{2^n-1} (-1)^{f(x)} \right|^2$$

which is equal to 1 if  $f(x)$  is constant and 0 if  $f(x)$  is balanced.

- 1.5) Measurement of the first  $n$  qubits yields  $|0\rangle^{\otimes n}$  with probability 1 if  $f(x)$  is constant; if  $f(x)$  is balanced, the probability is 0.

#### 2) Approach without using auxiliary qubit:

- 2.1) The quantum system is prepared in an  $n$  qubit state:

$$|\psi_0\rangle = |0\rangle^{\otimes n} \quad (8)$$

- 2.2) Each qubit undergoes a Hadamard transformation:

$$\begin{aligned} |\psi_1\rangle &= H^{\otimes n}|\psi_0\rangle \\ &= \left( \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle \right) \end{aligned} \quad (9)$$

- 2.3) The function  $f$  is encoded into a quantum oracle, which applies a phase shift conditioned on the function's output.

$$U_f|x\rangle = (-1)^{f(x)}|x\rangle \quad (10)$$

So,

$$U_f|\psi_1\rangle = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} (-1)^{f(x)}|x\rangle \quad (11)$$

- 2.4) Hadamard gates are applied to all qubits:

$$\begin{aligned} |\psi_2\rangle &= H^{\otimes n}|\psi_1\rangle \\ &= \frac{1}{2^n} \sum_{z=0}^{2^n-1} \left( \sum_{x=0}^{2^n-1} (-1)^{x \cdot z + f(x)} \right) |z\rangle \end{aligned} \quad (12)$$

From (12) the probability of measuring  $z = 0$ , corresponding to  $|0\rangle^{\otimes n}$ , is

$$\left| \frac{1}{2^n} \sum_{x=0}^{2^n-1} (-1)^{f(x)} \right|^2$$

which is equal to 1 if  $f(x)$  is constant and 0 if  $f(x)$  is balanced.

- 2.5) Measurement of qubits yields  $|0\rangle^{\otimes n}$  with probability 1 if  $f(x)$  is constant; if  $f(x)$  is balanced, the probability is 0.

#### IV. TWO-QUBIT DEUTSCH JOZSA

In this section, we present the quantum walk scheme for two-qubit Deutsch-Jozsa algorithm with auxiliary qubit and without using auxiliary qubit along with a discussion on the implementation of both schemes in a photonic system. For the two-bit case, the function  $f(x_1, x_2)$  has eight possible realizations as shown in Table I. In the standard circuit model the oracle (5) is achieved utilizing a sequence of  $X$  and CNOT gates applied to the auxiliary qubit. The choice and configuration of these gates depend on the specific function being implemented, ensuring that the oracle correctly encodes the function's behavior. The detailed oracle implementations for each function are presented in Fig. 1.

##### A. Quantum walk scheme using auxiliary qubit

The quantum system is initialized in a three-qubit state i.e., the particle undergoes a quantum walk on an closed graph consisting of four vertices where the coin state represents the auxiliary qubit and position space  $\text{span}\{|00\rangle, |01\rangle, |11\rangle, |10\rangle\}$  represents two working qubits. Initially, the particle starts

TABLE I: Inputs and outputs of all possible two-bit Boolean functions that are either constant or balanced. Functions (i) and (ii) correspond to constant functions, while functions (iii) to (viii) represent balanced functions. Specifically,  $f(x_1, x_2)$  is defined as: (i) 0, (ii) 1, (iii)  $x_1$ , (iv)  $x_2$ , (v)  $\overline{x_1}$ , (vi)  $\overline{x_2}$ , (vii)  $x_1 \oplus x_2$ , and (viii)  $\overline{x_1} \oplus x_2$ .

Inputs ( $x_1, x_2$ )	Functions $f(x_1, x_2)$							
	(i)	(ii)	(iii)	(iv)	(v)	(vi)	(vii)	(viii)
(0, 0)	0	1	0	0	1	1	0	1
(0, 1)	0	1	0	1	1	0	1	0
(1, 0)	0	1	1	0	0	1	1	0
(1, 1)	0	1	1	1	0	0	0	1

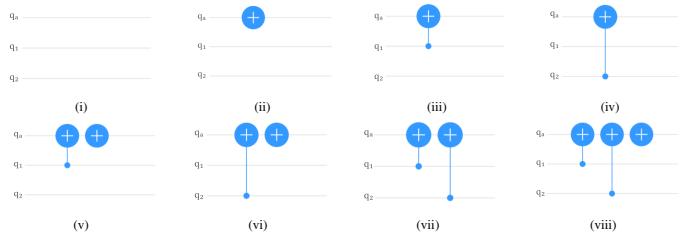


Fig. 1: Oracle implementations for all two-bit Boolean functions that are either constant or balanced given in Table I, by using  $X$  and CNOT gates in standard circuit model with auxiliary qubit [12]. The  $q_3$ ,  $q_1$ , and  $q_2$  represents auxiliary, first working and second working qubit, respectively.

with  $|0\rangle$  coin state and  $|00\rangle$  position state, after that coin operation  $\hat{C}(-\pi/2, 0, \pi/2, \pi/2)$  (1) is applied on coin (particle) state followed by the identity operation on position space to obtain the coin state  $|1\rangle$ . Both Hadamard transformations, as outlined in (4) and (7), are realized through a sequence of shift and coin operations. Specifically, the Hadamard gate on the auxiliary qubit (coin state) is implemented via the coin operation  $\hat{C}(-\pi/2, -\pi/2, \pi/2, \pi/4)$  (1) applied to the coin (particle) state, followed by an identity operation on the position space. The Hadamard gates acting on the working qubits in position space are implemented by evolving the coin state into a superposition across the position space. This is achieved through appropriate combinations of Pauli-X, Z, and Hadamard operations on the coin space, along with shift operations  $\{\hat{S}_+^a, \hat{S}_-^b\}$  (2). The detailed mathematical expressions for these operations can be found in [10]. The quantum walk based oracle implementations for all two-bit boolean functions that are either constant or balanced with auxiliary qubit (5) can be achieved by using a position-dependent coin operation  $\hat{C}(-\pi/2, 0, \pi/2, \pi/2)$  together with an identity shift operator i.e.,  $\sigma_x = \hat{C}(-\pi/2, 0, \pi/2, \pi/2) \otimes \hat{I}$  as shown in Fig. 2. The operations for the oracle implementation shown in Fig. 2 can be implemented in the photonic system using HWPs at angle  $\pi/4$  as shown in Fig. 3. The corresponding Jones matrices for the HWP rotated by angle  $\alpha$  w.r.t horizontal axis is given by

$$\text{HWP}(\alpha) = \begin{pmatrix} \cos 2\alpha & \sin 2\alpha \\ \sin 2\alpha & -\cos 2\alpha \end{pmatrix}. \quad (13)$$

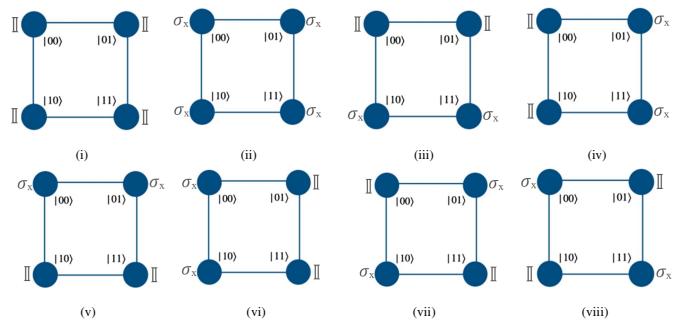


Fig. 2: Schematic illustration of the quantum walk based oracle implementations with auxiliary qubit for all two-bit boolean functions that are either constant or balanced given in Table I. Here,  $\mathbb{I}$  represents the identity coin operator together with an identity shift operator  $(\hat{C}(0, 0, 0, 0) \otimes \hat{I})$ .

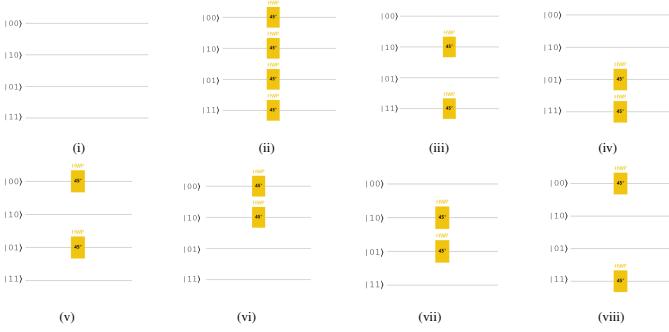


Fig. 3: Schematic illustration of the photonic implementation of the operations shown in Fig. 2. Here, the four path degrees ( $|00\rangle$ ,  $|10\rangle$ ,  $|01\rangle$ , and  $|11\rangle$ ) of freedom represent two working qubits and the polarization state of the photon represents the auxiliary qubit.

In the implementation of this scheme using photonic quantum walk, a photon undergoes a quantum walk within a four-path (mode) photonic circuit where the polarization state of the photon represents the auxiliary qubit and four path degrees of freedom represent two working qubits. The photon starts with  $|H\rangle$  polarization state in the 1st path ( $|00\rangle$ ), after that  $HWP(\pi/4)$  as in (13) is applied on the 1st path ( $|00\rangle$ ) to get  $|V\rangle$  polarization state. Both Hadamard transformations, as described in (4) and (7), are implemented using half-wave plates (HWPs) set at  $\pi/8$  ( $HWP(\pi/8)$ ) (13), 50:50 beam splitters (BSs), and mode permuters. The mode permuter facilitates the manipulation and reordering of photon paths, while the 50:50 beam splitter applies the Hadamard transformation to the photon's path degree of freedom. The transformation matrix for a 50:50 beam splitter, essential for generating superposition states, is given by

$$BS = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}. \quad (14)$$

The Hadamard transformations, as described in (4) and (7), are highlighted in (b) and (d) boxes in Fig. 4. The sequence of operations for the implementation of oracles are depicted in Fig. 3. The photonic circuit for the two-qubit photonic quantum walk-based Deutsch-Jozsa algorithm using an auxiliary qubit for the function  $f(x_1, x_2) = 0$  is shown in Fig. 4, where the (c) box represents the oracle. For the other functions listed in Table I, this box is replaced with the corresponding photonic circuit that implements the oracle for the given function, as shown in Fig. 3.

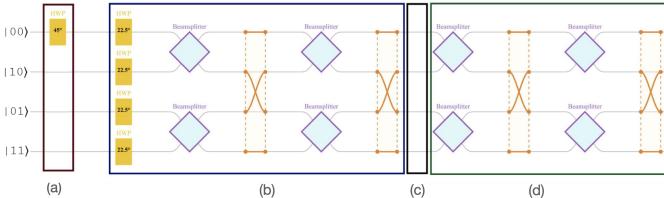


Fig. 4: The photonic circuit for the photonic quantum walk based two-qubit Deutsch-Jozsa algorithm using auxiliary qubit for the function ( $f(x_1, x_2) = 0$ ). The four path degrees ( $|00\rangle$ ,  $|01\rangle$ ,  $|11\rangle$ , and  $|10\rangle$ ) of freedom represent two working qubits and the polarization state represents the auxiliary qubit. Here, (a), (b), (c), and (d) boxes represent  $X$  gate on polarization state (auxiliary qubit),  $H$  gate on all the qubits, oracle, and  $H$  gate on two working qubits, respectively.

## B. Quantum walk scheme without using auxiliary qubit

The quantum system is initialized in a two-qubit state i.e., the particle undergoes a quantum walk on an open graph consisting of two vertices where the coin state represents the first qubit and position space  $span\{|0\rangle, |1\rangle\}$  represents the second qubit. Initially, the particle starts with  $|0\rangle$  coin state and  $|0\rangle$  position state. Both Hadamard transformations, as described in (11) and (12), are implemented through a sequence of shift and coin operations as discussed in Section IV-A. The quantum walk based oracle implementations for all two-bit boolean functions that are either constant or balanced without auxiliary qubit (10) can be achieved by applying identity operation  $\mathbb{I}$  and three position dependent evolution operators ( $\hat{O}_1$ ,  $\hat{O}_2$ ,  $\hat{O}_3$ ):

$$\begin{aligned} \hat{O}_1 &= \hat{C}\left(\frac{\pi}{2}, \frac{\pi}{2}, 0, \pi\right) \otimes \hat{I} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \otimes \hat{I}, \\ \hat{O}_2 &= \hat{C}\left(\frac{\pi}{2}, \frac{\pi}{2}, 0, 0\right) \otimes \hat{I} = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \otimes \hat{I}, \\ \hat{O}_3 &= e^{i\pi} \left( \hat{C}(0, 0, 0, 0) \otimes \hat{I} \right) = e^{i\pi} \left( \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \otimes \hat{I} \right), \\ \mathbb{I} &= \hat{C}(0, 0, 0, 0) \otimes \hat{I} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \otimes \hat{I}. \end{aligned} \quad (15)$$

The schematic illustration of the quantum walk based oracle implementations for all two-bit boolean functions that are either constant or balanced without using auxiliary qubit is shown in Fig 5. From (13) and (15),  $\hat{O}_1 = HWP(0)$  and  $\hat{O}_2 = HWP(\pi/2)$  so, the operations for the oracle implementation shown in Fig. 5 can be implemented in the photonic system a combination of phase shifters at an angle of  $\pi$  and half-wave plates (HWPs) at angles 0 and  $\pi$  as shown in Fig. 6 , where phase shifter at an angle of  $\pi$  applies a phase shift of  $e^{i\pi}$  to the mode where it is implemented i.e.,  $\hat{O}_3 = \text{PhaseShifter}(\pi)$ . In the photonic implementation, a single photon performs a quantum walk where its polarization encodes the first qubit and its two paths (modes) encode the second qubit. The photon starts with  $|H\rangle$  polarization state in the 1st path ( $|0\rangle$ ). Both Hadamard transformation, given by equations (9) and (12), are implemented using half-wave plates set at  $\pi/8$  ( $HWP(\pi/8)$ ) (13) and 50:50 beam splitters (14), as highlighted in (a) and (c) boxes in Fig. 7, respectively. The sequence of operations for the oracle implementation are depicted in Fig. 6. The photonic circuit for the two-qubit Deutsch-Jozsa algorithm based on photonic quantum walk without using auxiliary qubit for the function  $f(x_1, x_2) = 0$  is

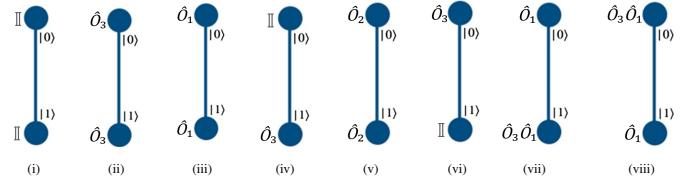


Fig. 5: Schematic illustration of the quantum walk based oracle implementations without auxiliary qubit for all two-bit boolean functions that are either constant or balanced given in Table I, by using identity operation  $\mathbb{I}$  and three position dependent evolution operators ( $\hat{O}_1$ ,  $\hat{O}_2$ ,  $\hat{O}_3$ ) as defined in (15).

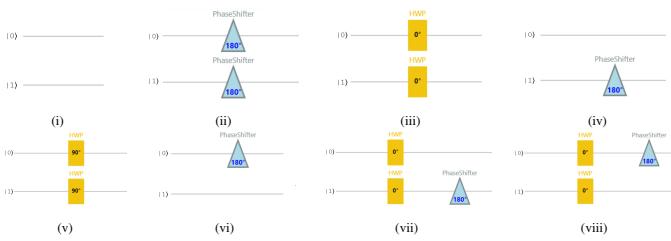


Fig. 6: Schematic illustration of the photonic implementation of the operations shown in Fig. 5. Here, the two path degrees ( $|0\rangle, |1\rangle$ ) of freedom represent second qubit and the polarization state of the photon represents the first qubit.

shown in Fig. 7, where the (c) box represents the oracle. For the other functions listed in Table I, this box is replaced with the corresponding photonic circuit that implements the oracle for the given function, as shown in Fig. 6.

#### V. BERNSTEIN-VAZIRANI ALGORITHM

The Bernstein-Vazirani algorithm is a quantum algorithm designed to determine a hidden binary string  $s$  using a single query to an oracle. This problem is formally defined as follows: input is an oracle that realizes the function  $f : \{0, 1\}^n \rightarrow \{0, 1\}$  of the form

$$f(x) = x \cdot s = x_1 s_1 \oplus x_2 s_2 \oplus \cdots \oplus x_n s_n, \quad (16)$$

where  $s$  is an unknown string of length  $n$ , the task is to determine  $s$  efficiently. Classically, determining  $s$  requires querying the function at least  $n$  times, each time using an input  $x$  that isolates one bit of  $s$ . However, the Bernstein-Vazirani algorithm determines  $s$  in a single query. The procedure is as follows:

- An  $n$ -qubit register is initialized in the state  $|\phi_0\rangle = |0\rangle^{\otimes n}$ , and a Hadamard transform  $H^{\otimes n}$  is applied to create an equal superposition:

$$|\phi_1\rangle = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle \quad (17)$$

- The function  $f(x)$  is embedded in an oracle operation  $U_f$ , which applies the transformation

$$U_f|x\rangle = (-1)^{f(x)}|x\rangle, \quad (18)$$

where  $f(x) = x \cdot s$ , this results in

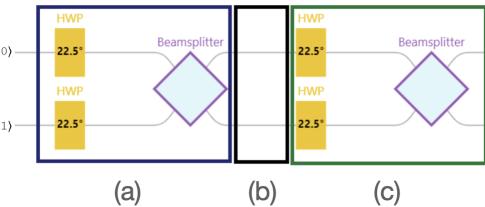


Fig. 7: The photonic circuit for the photonic quantum walk based two-qubit Deutsch-Jozsa algorithm without using auxiliary qubit for the function  $f(x_1, x_2) = 0$ . In this setup, the polarization state and the two path degrees of freedom ( $|0\rangle, |1\rangle$ ) correspond to the first and second qubits, respectively. The (a), (b), and (c) boxes denote the Hadamard ( $H$ ) transformation on both qubits, the oracle, and another Hadamard ( $H$ ) transformation on both qubits, respectively.

$$|\phi_2\rangle = U_f|\phi_1\rangle = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} (-1)^{x \cdot s} |x\rangle \quad (19)$$

- Applying another Hadamard transformation to each qubit,

$$\begin{aligned} |\phi_3\rangle &= H^{\otimes n}|\phi_2\rangle = \frac{1}{2^n} \sum_{y=0}^{2^n-1} \sum_{x=0}^{2^n-1} (-1)^{x \cdot s + x \cdot y} |y\rangle \\ &= \frac{1}{2^n} \sum_{y=0}^{2^n-1} \sum_{x=0}^{2^n-1} (-1)^{x \cdot (s \oplus y)} |y\rangle = |s\rangle \end{aligned} \quad (20)$$

- Measuring the quantum register ( all  $n$  qubits) determines the hidden string  $s$  with certainty, as the sum evaluates to 1 if  $s = y$  i.e.,  $s \oplus y = 0$ , and 0 otherwise.

The procedure described here does not utilize an auxiliary qubit; however, the same result can be achieved by incorporating an auxiliary qubit and applying the oracle transformation:

$$U_f|x\rangle|y\rangle = |x\rangle|y \oplus f(x)\rangle. \quad (21)$$

The Boolean function  $f(x) = x \cdot s$  (16) is a subset of constant and balanced functions. It is constant if  $s = 000\dots 0$  and balanced if  $s \neq 000\dots 0$ , as it then outputs 0 and 1 an equal number of times over all  $2^n$  inputs. Since the functions in the Bernstein-Vazirani algorithm form a subset of constant and balanced functions, the quantum walk scheme for the Bernstein-Vazirani algorithm, both with and without an auxiliary qubit, employs the same qubit encoding and Hadamard transformation as the corresponding quantum walk schemes for the Deutsch-Jozsa algorithm. Furthermore, for functions that belong to this subset, the oracles in the Bernstein-Vazirani algorithm (21), (18) are identical to those in the Deutsch-Jozsa algorithm (5), (10), utilizing the same set of quantum walk operations. For the Bernstein-Vazirani algorithm of two qubits, there are four possible hidden strings, as shown in Table II along with the functions associated with these strings. The oracle implementations in quantum walk schemes with and without an auxiliary qubit are shown in Fig. 2 and Fig. 5, while their implementations in a photonic system are presented in Fig. 3 and Fig. 6, where  $f(x_1, x_2)$  is defined as (i) 0, (iii)  $x_1$ , (iv)  $x_2$ , and (vii)  $x_1 \oplus x_2$ .

#### VI. DISCUSSION AND ANALYSIS

The quantum walk scheme for Deutsch-Jozsa algorithm presented in Section IV-B, which does not utilize an auxiliary qubit, is more resource-efficient than the scheme described in Section IV-A, as as it eliminates the need for an additional auxiliary qubit and reduces the total gates required, thus lowering the overall circuit complexity. The plot in Fig. 8 presents a comparison of the optical components required to

TABLE II: The four possible hidden strings and their corresponding functions in the two-bit Boolean function  $f(x) = x \cdot s$  (16).

Hidden String	Functions $f(x_1, x_2) =$
00	0
01	$x_2$
10	$x_1$
11	$x_1 \oplus x_2$

implement both schemes in the photonic system for all the two-bit Boolean functions listed in Table I. The component counts are determined based on the implementation of all gates, as depicted in Fig. 3, Fig. 4, Fig. 5, and Fig. 7. In this comparison, the mode-permuter is not considered a distinct physical component, as it can be naturally realized through path relabeling and the components essential for single-photon generation and measurement are also not included. For the quantum walk scheme for Deutsch-Jozsa algorithm using an auxiliary qubit,  $|00\rangle$  state of the working qubits corresponds to a photon present in the path state  $|00\rangle$ . Therefore, only a single-photon detector at the first path is required. For the quantum walk scheme for Deutsch-Jozsa algorithm without an auxiliary qubit, the state  $|00\rangle$  corresponds to a horizontally polarized ( $|H\rangle$ ) photon in path 1 ( $|0\rangle$ ). Thus, both polarization and path state need to be measured, which can be achieved using a PBS or a polarizer along with a single-photon detector. The plot in Fig. 8 also presents a comparison of the two-qubit Bernstein-Vazirani algorithm under both schemes (with and without an auxiliary qubit) for the four functions, where  $f(x_1, x_2)$  is defined as (i) 0, (iii)  $x_1$ , (iv)  $x_2$ , and (vii)  $x_1 \oplus x_2$ . In the measurement stage of the Bernstein-Vazirani algorithm, both working qubits must be measured in the scheme with an auxiliary qubit using single-photon detectors at all paths. In contrast, in the scheme without an auxiliary qubit, all qubits must be measured, which can be achieved using a PBS or a polarizer along with single-photon detectors at all the paths to measure both polarization and path states.

## VII. CONCLUSION

In this paper, we have presented the scheme for implementing the Deutsch-Jozsa and Bernstein-Vazirani algorithm using single-particle discrete-time quantum walk, both with and without an auxiliary qubit. We have given a set of experimentally realizable walk operations that can implement the oracles for the Deutsch-Jozsa and Bernstein-Vazirani algorithms along

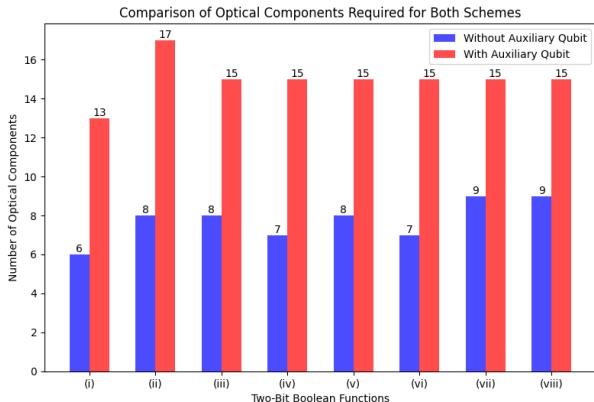


Fig. 8: The plot comparing the optical components required for implementing the quantum walk schemes of the Deutsch-Jozsa and Bernstein-Vazirani algorithms, both with and without an auxiliary qubit, in a photonic system. The comparison includes all two-bit Boolean functions listed in Table I for the Deutsch-Jozsa algorithm and the four two-bit boolean functions for the Bernstein-Vazirani algorithm defined as (i) 0, (iii)  $x_1$ , (iv)  $x_2$ , and (vii)  $x_1 \oplus x_2$ . The components essential for single-photon generation and measurement are not included in this comparison.

with the details to implement the operations in the photonic system. Detailed implementation of the oracles for two-qubit Deutsch-Jozsa and two-qubit Bernstein-Vazirani algorithm using this set of operations is presented in the paper, and this set of walk operations is sufficient to implement the oracles for the  $n$ -qubit Deutsch-Jozsa and Bernstein-Vazirani algorithms, as an appropriate combination of these operations enables the realization of any oracle for both algorithms. Unlike the standard circuit model, where oracle implementations rely on CNOT gates, the quantum walk approach enables a more straightforward realization using a simple set of quantum walk operations. A comprehensive analysis of the quantum walk schemes, with and without an auxiliary qubit in a photonic system, demonstrates that the scheme without an auxiliary qubit is more resource efficient, as it eliminates the need for an additional qubit and its associated quantum gates while also reducing the overall optical component count.

## ACKNOWLEDGMENT

This work utilized *Qniverse: A Unified Quantum Computing Platform* [12] for the construction of quantum circuits. This research was supported by the Ministry of Electronics and Information Technology (MeitY), Government of India, under the project titled “*HPC-based Quantum Accelerators for Enabling Quantum Computing on Supercomputers*”, Grant No. 4(3)/2022-ITEA.

## REFERENCES

- [1] M. A. Nielsen and I. L. Chuang, “Quantum Computation and Quantum Information,” Cambridge university press, 2010.
- [2] David Deutsch and Richard Jozsa, “Rapid Solution of Problems by Quantum Computation,” Proceedings of the Royal Society of London, Series A: Mathematical and Physical Sciences, 439(1907):553–558, 1992.
- [3] Ethan Bernstein and Umesh Vazirani, “Quantum Complexity Theory,” In Proceedings of the Twenty-Fifth Annual ACM Symposium on Theory of Computing, STOC ’93, pages 11–20, New York, NY, USA, 1993, Association for Computing Machinery.
- [4] Hao Li, Daowen Qiu, and Le Luo, “Distributed generalized deutsch-jozsa algorithm,” Computing and Combinatorics, pages 214–225, Singapore, 2025, Springer Nature Singapore.
- [5] Koji Nagata, Han Geurdes, Santanu Kumar Patro, Shahrokh Heidari, Ahmed Farouk, and Tadao Nakamura, “Generalization of the Bernstein-Vazirani Algorithm Beyond Qubit Systems,” Quantum Studies: Mathematics and Foundations, vol. 7, no. 1, pp. 17–21, 2020.
- [6] Abdeslem Khemakhmia and Habib Aissaoui, “Supersymmetric Extension of Deutsch-Jozsa Algorithm,” Quantum Information Processing, vol. 24, no. 5, p. 141, 2025.
- [7] Y. Aharonov, L. Davidovich, and N. Zagury, “Quantum Random Walks,” Phys.Rev. A, 48:1687–1690, Aug 1993.
- [8] Edward Farhi and Sam Gutmann, “Quantum Computation and Decision Trees,” Phys. Rev. A, 58:915–928, Aug 1998.
- [9] Salvador Elias Venegas-Andraca, “Quantum Walks: a comprehensive review,” Quantum Information Processing, 2012.
- [10] Shivani Singh, Prateek Chawla, Anupam Sarkar, and C. M. Chandrashekhar, “Universal quantum computing using Single-particle discrete-time quantum walk,” Scientific Reports, 2021.
- [11] Kanad Sengupta, K. Muhammed Shafi, S. P. Dinesh, Soumya Asokan, and C. M. Chandrashekhar, “Experimental realization of universal quantum gates and six-qubit state using photonic quantum walk,” 2024, arXiv:2403.06665.
- [12] Qniverse. <https://qniverse.in>