Quantum Information & Computation
ISSN 1533-7146 Volume 25 (2025) pp. 299–314
©Cerebration Science Publishing

Article

Reducing T-Count In Quantum String Matching Algorithm Using Relative-Phase Fredkin Gate

Byeongyong Park, Hansol Noh and Doyeol Ahn *

Department of Electrical and Computer Engineering and Center for Quantum Information Processing, University of Seoul, 163 Seoulsiripdae-ro, Dongdaemun-gu, Seoul 02504, Republic of Korea
* Corresponding author: dahn@uos.ac.kr

Received date: 25 October 2024; Accepted date: 2 June 2025; Published online: 10 July 2025

Abstract: The string-matching problem, ubiquitous in computer science, can significantly benefit from quantum algorithms due to their potential for greater efficiency compared to classical approaches. The practical implementation of the quantum string matching (QSM) algorithm requires fault-tolerant quantum computation due to the fragility of quantum information. A major obstacle in implementing fault-tolerant quantum computation is the high cost associated with executing T gates. This paper introduces the relative-phase Fredkin gate as a strategy to notably reduce the number of T gates (T-count) necessary for the QSM algorithm. This reduces the T-count from $14N^{3/2}\log_2 N - O(N^{3/2})$ to $8N^{3/2}\log_2 N - O(N^{3/2})$, where N represents the size of the database to be searched. Additionally, we demonstrate that our method is advantageous in terms of other circuit costs, such as the depth of T gates and the number of CNOT gates. This advancement contributes to the ongoing development of the QSM algorithm, paving the way for more efficient solutions in the field of computer science.

Keywords: quantum circuit; quantum string matching; T-count; relative phase Fredkin gate; Grover's search algorithm

1. Introduction

Grover's quantum search algorithm is renowned for providing a quadratic speedup in solving unstructured search problems [1,2]. A notable application of this speedup is in addressing the string-matching problem [3,4], where the goal is to find a pattern P of length M within a given string L of length N. This problem is fundamental in computer science due to its wide-ranging applications [5], including information retrieval [6], plagiarism detection [7], text mining [8], intrusion detection [9,10], language translation [11], and DNA sequence analysis [12–15], among others.

A significant challenge in implementing the quantum string matching (QSM) algorithm is the requirement for fault-tolerant quantum computation due to the inherent fragility of quantum information. In fault-tolerant quantum computations, circuits are constructed using a universal and fault-tolerant gate set. The Clifford + T gate set is commonly employed in various leading fault-tolerant schemes [16,17]. Within these approaches, Clifford gates are relatively straightforward to implement, often through transversal methods. In contrast, T gates are resource-intensive and require techniques like magic state distillation, which significantly increase the implementation costs [18–21]. Consequently, optimizing the number of T gates (T-count) is crucial for the efficient execution of large-scale quantum algorithms, including QSM.

Several studies have developed QSM algorithms [3,4,22], with the most notable contribution being the work of Niroula and Nam [4]. Their work is particularly significant because it provides an explicit quantum circuit implementation of the QSM algorithm. In their circuit design, they introduce a component called the "cyclic shift operator," which plays an essential role in constructing the Grover's operator specific to the QSM. The construction of these operators in the QSM circuit is the primary factor that increases the T-count within the QSM algorithm.

In this paper, we propose a method to reduce the T-count for the QSM circuit, based on the circuit design proposed by Niroula and Nam [4]. The QSM circuit in Niroula and Nam's work requires approximately $14N^{3/2}\log_2 N$ T gates, the majority of which arise from the synthesis of controlled-SWAP (Fredkin) gates in cyclic shift operators. They synthesized each Fredkin gate into Clifford + T gates using seven T gates (see Figure 1a). As part of our strategy

DOI: 10.2478/qic-2025-0016

to reduce the T-count, we introduce a new concept: the relative-phase Fredkin gate, inspired by the relative-phase Toffoli gate [23]. We synthesize a relative-phase Fredkin gate that contains only four T gates (see Figure 1b) and prove that each Fredkin gate in the QSM circuit can be replaced with any relative-phase Fredkin gate. As a result, we reduce the leading-order term of the T-count for QSM to $8N^{3/2}\log_2 N$. Additionally, we demonstrate that our approach provides advantages in terms of reducing other circuit costs, such as the depth of T gates (T-depth) and the number of CNOT gates (CNOT-count).

The circuit cost reduction presented in this work is achieved by incorporating relative-phase encoding into the quantum states of the QSM algorithm proposed by Niroula and Nam [4]. This approach is feasible because the relevant information about the string and pattern is encoded in basis states rather than amplitudes, preserving measurement outcomes performed in the computational basis. Furthermore, our relative-phase-based optimization strategy effectively reduces circuit costs not only for the QSM algorithm but also for a broader class of quantum algorithms, particularly those employing controlled quantum gates such as Fredkin or similar controlled permutations.

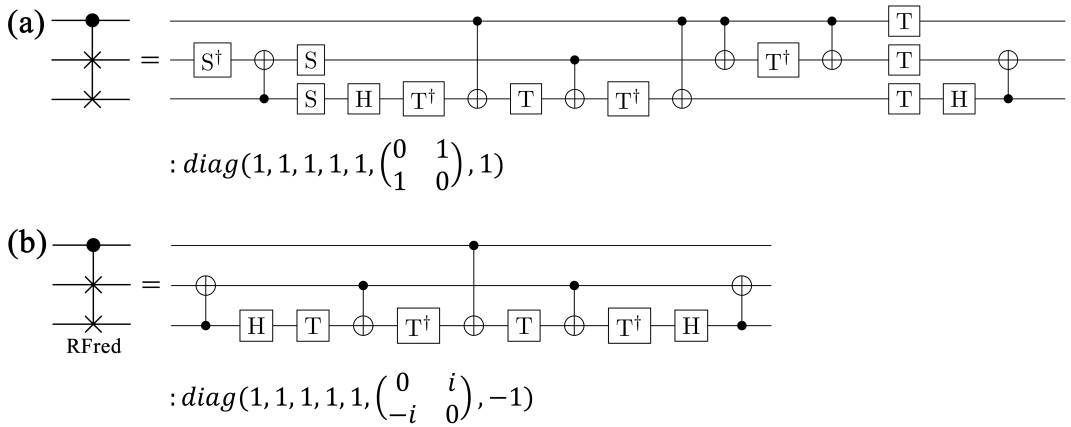

Figure 1. Decomposition of the Fredkin gate and the relative-phase Fredkin gate into the Clifford + T gate set. (**a**) Decomposition of the Fredkin gate used in Ref. [4], which requires seven T gates. (**b**) Decomposition of the relative-phase Fredkin gate used in this study, which requires four T gates. Implementing the QSM algorithm with the relative-phase Fredkin gate significantly reduces the T-count, a critical factor in minimizing space-time costs in fault-tolerant quantum computing.

2. Backgrounds

2.1. Grover's Algorithm

Grover's algorithm is a quantum algorithm that offers a quadratic speedup over classical search algorithms in unstructured search problems [1,2]. This section briefly explains Grover's algorithm using general quantum amplitude amplification [2].

Consider a Boolean function $f : \{0,1,\ldots,2^n-1\} \to \{0,1\}$. Within an n-qubit system, two subspaces, G and B, are defined as follows: $G \equiv \mathrm{span}(\{|x\rangle \mid f(x)=1\})$ and $B \equiv \mathrm{span}(\{|x\rangle \mid f(x)=0\})$. Note that $I_n = P_G + P_B$, where P_G and P_B are projectors onto G and B, respectively. Given a unitary operator A acting on the n-qubit system, $A|0\rangle_n \equiv |\psi\rangle$ can be expressed as a linear combination of $P_G|\psi\rangle$ and $P_B|\psi\rangle$ as shown in Eq. (1). Here, the states $|\phi_g\rangle$ and $|\phi_b\rangle$ are normalized states of $P_G|\psi\rangle$ and $P_B|\psi\rangle$, respectively.

$$A|0\rangle_n \equiv |\psi\rangle = P_G|\psi\rangle + P_B|\psi\rangle = \sin\theta|\phi_g\rangle + \cos\theta|\phi_b\rangle \tag{1}$$

In this study, we call the operator A the initialization operator.

The primary goal of Grover's algorithm is to find (or measure) the state in G with high probability. The operators utilized in Grover's algorithm are defined in Eqs. (2) to (5):

$$Q \equiv -R_\psi R_g \tag{2}$$

$$R_\psi \equiv I - 2|\psi\rangle\langle\psi| \equiv AR_0A^{-1} \tag{3}$$

$$R_0 \equiv I - |0\rangle_n\langle 0| \tag{4}$$

$$R_g \equiv I - 2|\phi_g\rangle\langle\phi_g| \tag{5}$$

We refer to operator Q as Grover's operator. The process of Grover's algorithm can be delineated as follows: First, an n-qubit is set to state $|0\rangle_n$, after which the operator A is applied to the state $|0\rangle_n$. Next, Q^k is applied to the state $A|0\rangle_n = |\psi\rangle$, where $k = \lfloor \pi/4\theta \rfloor$. Finally, a measurement is conducted on the n-qubit state. The post measurement state becomes a state in G with a probability higher than $\max(1 - \sin^2\theta, \sin^2\theta)$ [2].

2.2. *Quantum String Matching Algorithm*

This section describes the QSM algorithm and its circuit representation described in Ref. [4]. String matching aims to locate a pattern P of length M in a given string L of length N. For simplicity of analysis, we assume P and L to be binary digit strings and $N = 2^n$ throughout this study.

The core subcircuit of the algorithm is the cyclic shift operator C. The cyclic shift operator C on the $(n+N)$-qubit system is defined as Eq. (6), where $|k\rangle_n$ is a computational basis state, the additions are modulo N additions, and each x_i is a binary number 0 or 1.

$$C|k\rangle_n|x_0x_1x_2\ldots x_{N-1}\rangle_N = |k\rangle_n|x_{0+k}\,x_{1+k}\,x_{2+k}\ldots x_{N-1+k}\rangle_N \tag{6}$$

To construct the circuit of this operator, the cyclic shift operator C is divided into operators C_k, which are defined as follows.

$$C_k|l\rangle_n|x_0x_1x_2\ldots x_{N-1}\rangle_N = \begin{cases} |l\rangle_n|x_0x_1x_2\ldots x_{N-1}\rangle_N, & \text{if } l \neq k, \\ |l\rangle_n|x_{0+k}\,x_{1+k}\,x_{2+k}\ldots x_{N+k-1}\rangle_N, & \text{if } l = k. \end{cases} \tag{7}$$

where $|l\rangle_n$ is a computational basis state. Then, the cyclic shift operator C can then be rewritten as described in Eq. (8).

$$C = C_{2^0}C_{2^1}C_{2^2}\ldots C_{2^{n-1}} \tag{8}$$

Constructing a C_{2^j} operator requires a maximum of $N-1$ controlled-SWAP (Fredkin) gates. The process of constructing the C_{2^j} operator is divided into n stages. In the first stage, $N/2$ digits are moved to the right place using $N/2$ Fredkin gates. In the second stage, $N/4$ digits are moved to the right place using $N/4$ Fredkin gates, and so on. The total number of Fredkin gates used to construct the C_{2^j} operator is at most $N/2 + N/4 + \ldots + 1 = N-1$ (see Figure 2). Optionally, the Fredkin gates at each stage of constructing a C_{2^j} operator can be parallelized using $N/2-1$ ancilla qubits and fan-out operations with $N-2$ CNOT gates. The exact number of Fredkin gates required to construct the cyclic shift operator C can be found in Appendix A.

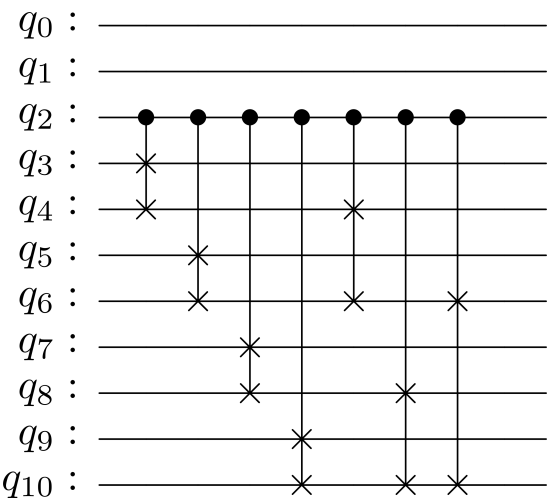

Figure 2. Construction of a C_{2^0} operator. In this figure, we show a C_{2^0} circuit when $N = 8$.

To implement Grover's algorithm, the initialization operator A such that $A|0\rangle_{(n+N+M)} = |\psi\rangle$ needs to be defined. The process of applying A to state $|0\rangle_{(n+N+M)}$ is as follows (see Figure 3a):

(i) Prepare three quantum registers with n, N, and M qubits. We call the n-qubit register the first register, N-qubit register the second register, and M-qubit register the third register. All the states were initialized to state $|0\rangle$.

(ii) Apply $H^{\otimes n}$ to the first register and apply X gates to encode L and P in the second and third registers, respectively. Then, the entire state becomes state $|\psi_1\rangle$ in Eq. (9). $|L\rangle_N$ and $|P\rangle_M$ in Eq. (9) are defined in Eqs. (10) and (11), respectively, where each l_j and p_j is a binary number.

$$|\psi_1\rangle = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} |k\rangle_n |L\rangle_N |P\rangle_M \tag{9}$$

$$|L\rangle_N = |l_0 l_1 l_2 \dots l_{N-1}\rangle \tag{10}$$

$$|P\rangle_M = |p_0 p_1 p_2 \dots p_{M-1}\rangle \tag{11}$$

(iii) Apply cyclic shift operator C to the first and second registers. Subsequently, the state evolves into state $|\psi_2\rangle$ in Eq. (12), where the additions are modulo N additions.

$$|\psi_2\rangle = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} |k\rangle_n |l_{0+k} l_{1+k} l_{2+k} \dots l_{N-1+k}\rangle |p_0 p_1 p_2 \dots p_{M-1}\rangle \tag{12}$$

(iv) Apply M CNOT gates to the first M qubits in the second register and to all qubits in the third register. For the CNOT operations, each qubit in the second register serves as the control qubit, and each qubit in the third register serves as the target qubit (see Figure 3b). Subsequently, the state evolves into state $|\psi\rangle$ as described in Eq. (13).

$$\begin{aligned} |\psi\rangle = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} |k\rangle_n |l_{0+k} l_{1+k} \dots l_{N+k-1}\rangle \\ |(p_0 \oplus l_{0+k})(p_1 \oplus l_{1+k}) \dots (p_{M-1+k} \oplus l_{M-1+k})\rangle \end{aligned} \tag{13}$$

In the remainder of this study, we use the notation in Eqs. (14) and (15).

$$|l_{0+k} l_{1+k} \dots l_{N+k-1}\rangle \equiv |L_k\rangle_N \tag{14}$$

$$|(p_0 \oplus l_{0+k})(p_1 \oplus l_{1+k}) \dots (p_{M-1+k} \oplus l_{M-1+k})\rangle \equiv |P_k\rangle_M \tag{15}$$

We define the state in G as the state in which the third register is in state $|0\rangle_M$.

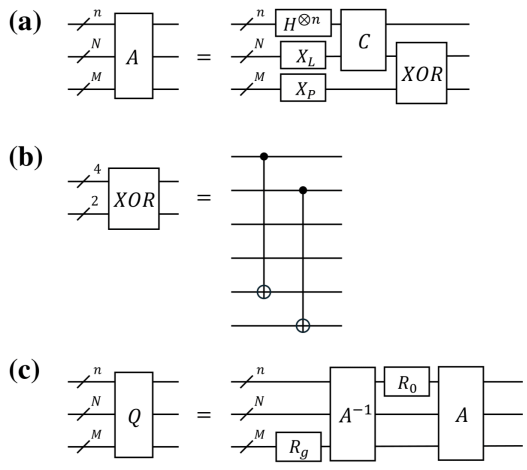

Figure 3. Construction of Grover's operator $Q = AR_0A^{-1}R_g$ for QSM algorithm. (**a**) Structure of the initialization operator A. X_L and X_P represent X gates to encode L and P, respectively. C represents the cyclic shift operator. XOR transforms state $|\psi_2\rangle$ in Eq. (12) into state $|\psi\rangle$ in Eq. (13). (**b**) Example of constructing an XOR operation when $N = 4$ and $M = 2$. (**c**) Structure of Grover's operator $Q = AR_0A^{-1}R_g$ for the QSM algorithm. Operators R_0 and R_g can be constructed using multi-qubit controlled-Z gates and single-qubit gates. Repeated application of Grover's operator amplifies the amplitude of the desired matching states.

As described earlier, Grover's operator Q is given by $AR_0A^{-1}R_g$ (see Figure 3c). If amplitude amplification is carried out using Grover's operator $Q = AR_0A^{-1}R_g$ on the $n+N+M$ qubits, with operator A defined as in Figure 3a, the pattern P can be identified within the given string L. R_0 and R_g can be constructed using X gates and a multi-qubit controlled-Z gate. A k-qubit controlled-Z gate is decomposed using two H gates and a k-qubit Toffoli gate. In Ref. [4], multi-qubit Toffoli gates are decomposed using relative-phase Toffoli gates in Ref. [23]. The calculations regarding the size and cost of R_0, which is not explicitly presented in Ref. [4], can be found in Appendix B.

3. Proposed QSM Circuit: Design and Validation

First, we define the relative-phase Fredkin gate analogously to the relative-phase Toffoli gate [23]. The Fredkin gate, also known as the controlled-SWAP gate, has a matrix representation on a computational basis given by $\mathrm{diag}(1,1,1,1,1,\left[\begin{smallmatrix}0&1\\1&0\end{smallmatrix}\right],1)$. We extend this to define the relative-phase Fredkin gate, which has a matrix representation of $\mathrm{diag}(z_0,z_1,z_2,z_3,z_4,\left[\begin{smallmatrix}0&z_6\\z_5&0\end{smallmatrix}\right],z_7)$, where each z_i is an arbitrary complex number with norm 1.

A Fredkin gate can be constructed using a Toffoli gate and two CNOT gates (see Figure 4a) [24]. Similarly, a relative-phase Fredkin gate can be constructed using a relative-phase Toffoli gate and two CNOT gates (see Figure 4b). This construction can be easily demonstrated through a straightforward matrix multiplication:

$$\begin{aligned}
&\mathrm{diag}\left(\begin{bmatrix}1&0&0&0\\0&0&0&1\\0&0&1&0\\0&1&0&0\end{bmatrix},\begin{bmatrix}1&0&0&0\\0&0&0&1\\0&0&1&0\\0&1&0&0\end{bmatrix}\right)\cdot\mathrm{diag}(z_0,z_1,z_2,z_3,z_4,z_5,\begin{bmatrix}0&z_7\\z_6&0\end{bmatrix})\cdot\\
&\mathrm{diag}\left(\begin{bmatrix}1&0&0&0\\0&0&0&1\\0&0&1&0\\0&1&0&0\end{bmatrix},\begin{bmatrix}1&0&0&0\\0&0&0&1\\0&0&1&0\\0&1&0&0\end{bmatrix}\right)=\mathrm{diag}(z_0,z_1,z_2,z_3,z_4,\begin{bmatrix}0&z_6\\z_5&0\end{bmatrix},z_7)
\end{aligned}\tag{16}$$

Using this method, as shown in Figure 1b, we constructed a relative-phase Fredkin gate from the relative-phase Toffoli gate described in Ref. [23]. The relative-phase Fredkin gate in Figure 1b has a matrix representation of $\mathrm{diag}(1,1,1,1,1,\left[\begin{smallmatrix}0&i\\-i&0\end{smallmatrix}\right],-1)$.

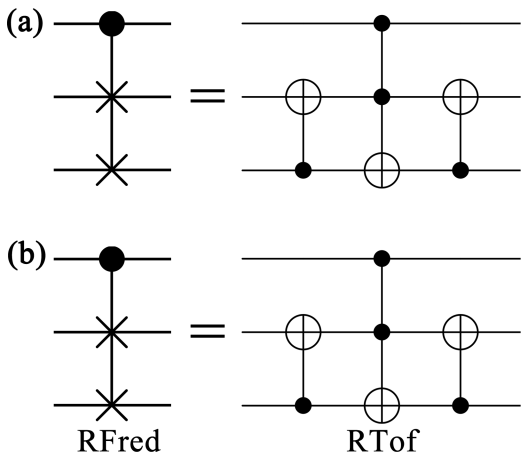

Figure 4. Decomposition of Fredkin gate and relative-phase Fredkin gate. (**a**) Construction of a Fredkin gate using a Toffoli gate and two CNOT gates reported in Ref. [24]. (**b**) Construction of a relative-phase Fredkin gate using a relative-phase Toffoli gate and two CNOT gates.

Next, we prove that any relative-phase Fredkin gate can replace the Fredkin gates used to construct the cyclic shift operator C in the QSM algorithm.

Lemma 1. *Any quantum circuit X_r composed solely of relative-phase Fredkin gates can be represented by Eq. (17), where X is a quantum circuit in which all relative-phase Fredkin gates in X_r are replaced with Fredkin gates, and D_r and D_l are circuits with diagonal matrix representation.*

$$X_r = X \cdot D_r = D_l \cdot X \tag{17}$$

Proof. It is sufficient to verify Eq. (18), where RFred represents an arbitrary relative-phase Fredkin gate, Fred represents a Fredkin gate, and D_1 and D_2 represent circuits with a diagonal matrix representation.

$$\text{RFred} = \text{Fred} \cdot D_1 = D_2 \cdot \text{Fred} \tag{18}$$

Basic matrix multiplication can prove Eq. (18):

$$\begin{aligned} &\text{diag}\left(z_0, z_1, z_2, z_3, z_4, \begin{bmatrix} 0 & z_6 \\ z_5 & 0 \end{bmatrix}, z_7\right) \\ &= \text{diag}(1,1,1,1,1,\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix},1) \cdot \text{diag}(z_0, z_1, z_2, z_3, z_4, z_5, z_6, z_7) \\ &= \text{diag}(z_0, z_1, z_2, z_3, z_4, z_6, z_5, z_7) \cdot \text{diag}(1,1,1,1,1,\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix},1) \end{aligned} \tag{19}$$

□

Theorem 1. *Even if arbitrary relative-phase Fredkin gates replace the Fredkin gates in constructing the cyclic shift operator C in the QSM algorithm described in Ref. [4], the algorithm yields the same results.*

Proof. Assume that we replace all Fredkin gates in cyclic shift operator C with relative-phase Fredkin gates. We express operators A and C with relative-phase Fredkin gates as A' and C'. By Lemma 1, C' can be written as $C \cdot D$, where D is an $(n+N)$-qubit operator with a diagonal matrix representation. Subsequently, $A|0\rangle_n|0\rangle_N|0\rangle_M$ and $A'|0\rangle_n|0\rangle_N|0\rangle_M$ evolve into the states in Eqs. (20) and (21), where each z_k is a complex number with norm 1.

$$A|0\rangle_n|0\rangle_N|0\rangle_M = |\psi\rangle = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} |k\rangle_n|L_k\rangle_N|P_k\rangle_M \equiv \sin\theta|\phi_g\rangle + \cos\theta|\phi_b\rangle \tag{20}$$

$$A'|0\rangle_n|0\rangle_N|0\rangle_M = |\psi'\rangle = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} z_k|k\rangle_n|L_k\rangle_N|P_k\rangle_M \equiv \sin\theta'|\phi'_g\rangle + \cos\theta|\phi'_b\rangle \tag{21}$$

Note that $\theta = \theta'$ because states $|k\rangle_n|L_k\rangle_N|P_k\rangle_M$ with different k values are orthonormal. Therefore, Grover's algorithm using A' gives the same results as the case using A. □

Based on Theorem 1, we can replace every Fredkin gate in the QSM algorithm with the relative-phase Fredkin gate of Figure 1b. Seven T gates are required to synthesize a Fredkin gate into Clifford + T gates (see Figure 1a). However, only four T gates are required to synthesize the relative-phase Fredkin gate of Figure 1b. Therefore, with this replacement, the cyclic shift operator C' is decomposed into Clifford + T gates using $4(N\log_2 N - N + 1)$ T gates. This results in reducing the leading-order term of the T-count in the QSM algorithm from $14N^{3/2}\log_2 N$ to $8N^{3/2}\log_2 N$, assuming that $N^{1/2}$ Grover iterations (the number of executions of the Grover operator) are implemented.

To validate the proposed QSM circuit, we simulated the QSM algorithm using the quantum simulator in Qiskit [25]. As shown in Figure 5, the proof-of-principle results exhibit excellent agreement with the expected outcomes of Grover's search algorithm. Our designed circuit targeted the pattern "11" within the data sequence "00110000". We varied the number of Grover iterations from 0 to 9 and compared the probabilities of finding the correct answer between the theoretically ideal Grover's search algorithm and the QSM algorithm implemented with the relative-phase Fredkin gate. Each QSM simulation was executed 10,000 times to estimate the probabilities.

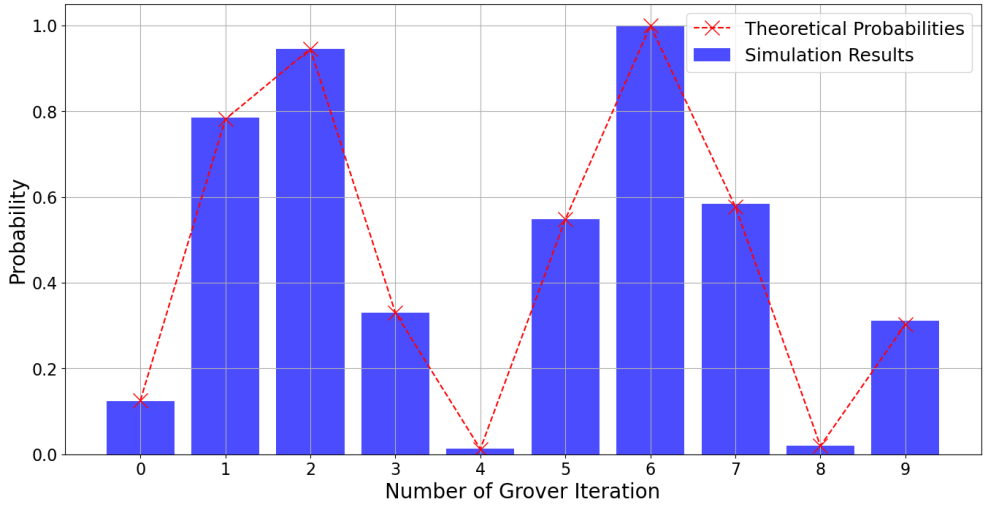

Figure 5. Comparison of the theoretical probabilities of Grover's search and the results of the QSM simulation with relative-phase Fredkin gates. Each QSM simulation was executed 10,000 times to estimate the probabilities. The designed circuit was intended to identify the pattern "11" within the data sequence "00110000". The x-axis shows the number of Grover iterations, and the y-axis displays the corresponding probabilities. The red line plot with "x" markers represents the theoretical probabilities of Grover's search algorithm functioning correctly, while the blue bar graph indicates the simulation results of the QSM algorithm with relative-phase Fredkin gates. The simulation data aligns closely with the theoretical expectations, demonstrating the correctness of the QSM algorithm with relative-phase Fredkin gates.

4. Results

In this section, we present the T-count required for the implementation of the QSM algorithm with relative-phase Fredkin gates. When Grover's operator is repeatedly implemented $N^{1/2}$ times, the leading order term of the total T-count for the QSM algorithm is reduced from $14N^{3/2}\log_2 N$ to $8N^{3/2}\log_2 N$. Additionally, we observe further T-count reductions in each iteration of $A'R_0(A')^{-1}$, where the primed operators refer to those using the relative-phase Fredkin gates. To construct C', we sequentially combine $C'_1, C'_2, C'_4, \ldots$ in sequence. For $(C')^{-1}$, we combine the inverse gates of the gates constituting C' in reverse order. Then, as shown in Figure 6, there is a T-count reduction around R_0. Each time the structure of $A'R_0(A')^{-1}$ appears in the QSM circuit, the T-count is reduced by $2N$ compared to the case of the QSM algorithm with Fredkin gates.

The introduction of the proposed relative-phase Fredkin gate for the QSM algorithm also benefits the T-depth by allowing the parallelization of the T gates without additional ancilla qubits for the construction of the cyclic shift operator. The QSM algorithm described in Ref. [4] parallelizes Fredkin gates in the cyclic shift operator by using $N/2-1$ ancilla qubits and fan-out operations with CNOT gates, which leads to the parallelization of the T gates. In contrast, the use of relative-phase Fredkin gates allows for automatic parallelization of T gates without the need for fan-out operations or additional ancilla qubits, as depicted in Figure 7. Compared to the Fredkin gate in Figure 1a with a T-depth of five, the relative-phase Fredkin gate in Figure 1b has a T-depth of four. Consequently, the T-depth required for the cyclic shift operator is reduced to $4/5$ of the result in Ref. [4] without additional ancilla qubits. The use of relative-phase Fredkin gates also offers advantages in terms of other circuit costs, such as the number of CNOT gates. In Table 1, the circuit cost of the QSM algorithms proposed in this paper is compared against those previously reported.

One might argue that the presence of an $O(N^{3/2})$ term in the second dominant component of the T-count, appearing in both our result and Ref. [4], could diminish the significance of the improvement achieved in the leading coefficient. However, the difference in T-count between the two approaches is given by $N^{3/2}(6\log_2 N-4)+O(N\log N)$, which remains positive even for relatively small values of N and continues to grow as the problem size increases. As illustrated in Figure 8, the T-count difference between our proposed approach and the one in Ref. [4] becomes increasingly significant with larger values of N. Given that the QSM algorithm is expected to exhibit quantum advantage primarily for large-scale instances, this trend indicates that the proposed method offers progressively greater benefits as the input size grows.

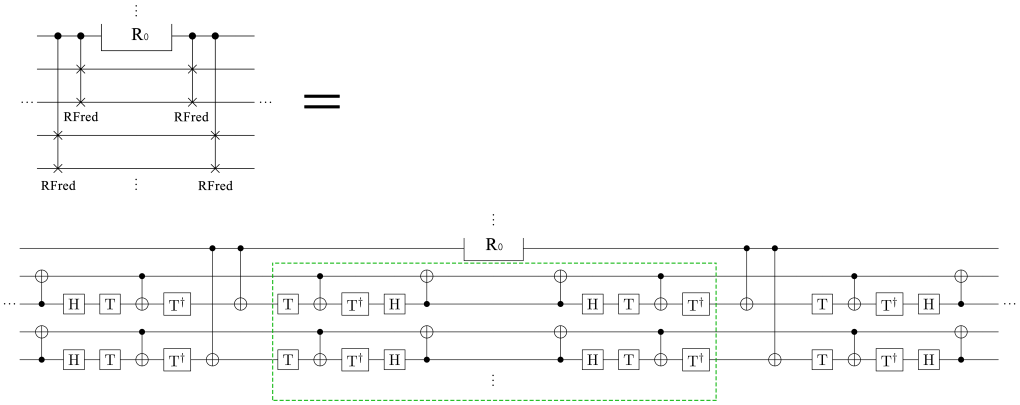

Figure 6. Circuit of $A'R_0(A')^{-1}$. The gates within the dashed green box are cancelled out, resulting in a reduction of $2N$ T-count.

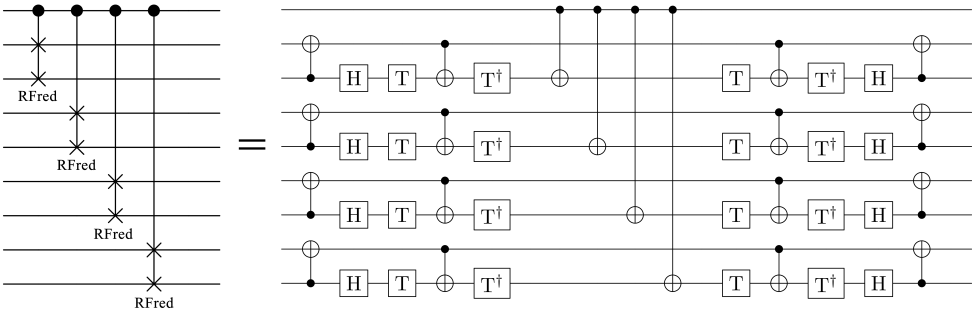

Figure 7. Automatic parallelization of T gates in constructing the cyclic shift operator when using the relative-phase Fredkin gate shown in Figure 1b.

Table 1. The table presents the required resources for the QSM algorithms, including T-count (the number of T gates), T-depth (the depth of T gates), CNOT-count (the number of CNOT gates), and Qubit-count (the number of qubits). Here, N represents the size of the database to be searched, and M represents the size of the pattern. For the comparison, we assume that $N^{1/2}$ Grover iterations are implemented.

Source	Ref. [4]	This Work
T-count	$14N^{3/2}\log_2 N - 14N^{3/2} + 7N\log_2 N - 7N + 8N^{1/2}\log_2 N + N^{1/2}(8M-20) + 7 = 14N^{3/2}\log_2 N - O(N^{3/2})$	$8N^{3/2}\log_2 N - 10N^{3/2} + 4N\log_2 N - 4N + 8N^{1/2}\log_2 N + N^{1/2}(8M-26) + 1 = 8N^{3/2}\log_2 N - O(N^{3/2})$
T-depth	$5N^{1/2}\log_2^2 N + 9N^{1/2}\log_2 N + N^{1/2}(4M+2) + \frac{5}{2}\log_2^2 N + \frac{5}{2}\log_2 N = 5N^{1/2}\log_2^2 N + O(N^{1/2}\log N)$	$4N^{1/2}\log_2^2 N + 8N^{1/2}\log_2 N + N^{1/2}(4M-2) + 2\log_2^2 N + 2\log_2 N = 4N^{1/2}\log_2^2 N + O(N^{1/2}\log N)$
CNOT-count	$16N^{3/2}\log_2 N - 14N^{3/2} + 7N\log_2 N - 7N + 10N^{1/2}\log_2 N + N^{1/2}(8M-10) + M + 7 = 16N^{3/2}\log_2 N - O(N^{3/2})$	$10N^{3/2}\log_2 N - 10N^{3/2} + 5N\log_2 N - 5N + 6N^{1/2}\log_2 N + N^{1/2}(8M-14) + M + 5 = 10N^{3/2}\log_2 N - O(N^{3/2})$
Qubit-count	$\frac{3}{2}N + \log_2 N + M - 1$	$N + \log_2 N + M$

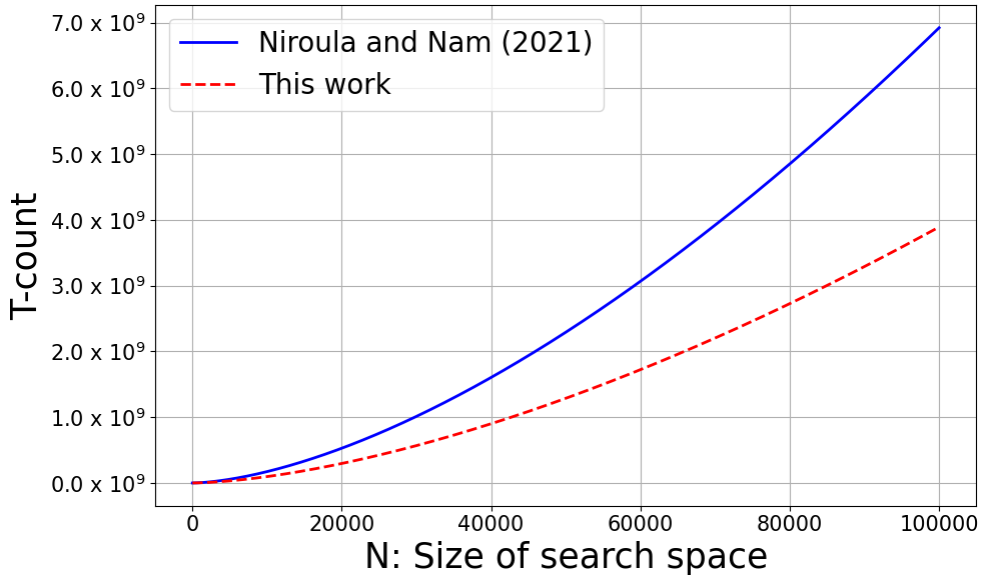

Figure 8. Comparison of the T-counts for the QSM algorithm. The red dashed line indicates the T-count derived in this study, while the blue solid line corresponds to the T-count reported in Ref. [4]. For consistency in comparison, the pattern size M is fixed at 100.

5. Discussion

The string-matching problem is a widespread challenge in computer science. As the scale of the problem increases, it becomes essential to enhance the efficiency of algorithms by reducing computational costs. In this paper, we have reduced the T-count of the QSM algorithm from $14N^{3/2}\log_2 N - O(N^{3/2})$ to $8N^{3/2}\log_2 N - O(N^{3/2})$. Additionally, our approach reduces other circuit costs of the QSM algorithm, including T-depth. Given that T gates dominate the cost of implementing fault-tolerant quantum circuits, our results may expedite the utilization of quantum advantages in solving string-matching problems, potentially impacting fields such as bioinformatics, text processing, and cryptography.

Our primary strategy for reducing circuit costs is to incorporate relative-phase encoding into the quantum states used in the QSM algorithm presented in Ref. [4]. This approach is feasible because the QSM algorithm from Ref. [4] stores information about the string and pattern in basis states rather than in quantum amplitudes. We anticipate that this circuit cost reduction strategy could be broadly applicable to various quantum algorithms. A practical example involving quantum image processing is provided in Appendix C.

To facilitate relative-phase encoding, we introduced the concept of the relative-phase Fredkin gate, inspired by the relative-phase Toffoli gate [23]. Given the extensive use of relative-phase Toffoli gates in optimizing quantum circuits [23,26–28], we expect relative-phase Fredkin gates to have similarly broad applications in quantum circuit optimization. An illustrative example of this application is presented in Appendix D.

By employing the Toffoli gate implementation method presented in Ref. [29] and the Fredkin gate synthesis method presented in Ref. [24], it is possible to execute a Fredkin gate using four T gates. This method can achieve a T-count reduction comparable to our approach. However, this Fredkin gate implementation method necessitates measurement, classically controlled gates, an ancilla qubit, and additional Clifford gates. In contrast, our method, which utilizes the relative-phase Fredkin gate shown in Figure 1b, employs a straightforward circuit synthesis approach. This enables us to reduce the T-count without incurring additional circuit costs. One might also argue that using the method from Ref. [26] allows a Fredkin gate to be implemented with a single T-depth, thereby further reducing the T-depth of the QSM algorithm. However, while this method can create a single Fredkin gate with a lower T-depth, the T gates of different Fredkin gates are not parallelized, ultimately resulting in a higher T-depth compared to our method.

Recently, K. Prousalis et al. proposed a quantum string-matching algorithm based on quantum image processing techniques, where string-matching is performed using a quantum-generated dot-matrix image followed by quantum median filtering to detect similarity patterns [30]. Their approach achieves a time complexity of $O(N^2)$ and a space complexity of $O(\log N)$ for a string of length N, demonstrating significant improvements in memory efficiency compared to previous quantum methods. They utilized the NEQR (novel enhanced quantum representation) framework

[31], which encodes image information in basis states rather than in quantum amplitudes. Therefore, our approach, which incorporates relative-phase encoding in quantum states, may also be applicable to the NEQR-based QSM algorithm proposed by Prousalis et al. [30], potentially reducing quantum circuit complexity in their framework as well. Exploring this possibility would be an interesting direction for future research.

Author Contributions
Byeongyong Park: Conceptualization, Methodology, Software, Formal Analysis, Investigation, Visualization, Writing—Original Draft, Writing—Review and Editing. Hansol Noh: Writing—Review and Editing. Doyeol Ahn: Supervision, Funding Acquisition, Writing—Review and Editing. All authors have read and agreed to the published version of the manuscript.

Funding
This work is supported by a 2024 Research grant from the University of Seoul.

Conflicts of Interest
Author declares no conflict of interest.

Data Availability Statement
The data that support the findings of this study are available from the corresponding author upon reasonable request.

Acknowledgments
We acknowledge support from a 2024 Research grant from the University of Seoul.

Appendix A

In this appendix, we present the exact number of Fredkin gates required to construct the cyclic shift operator C in the QSM algorithm.

Lemma A1. *When the size of the data to be searched is $N = 2^n$, constructing $C_{(2^0)}$ requires $2^n - 1$ Fredkin gates.*

Proof. It is sufficient to prove that a binary string $(a_0, a_1, a_2, \ldots, a_{(2^n-1)})$ can be transformed into $(a_1, a_2, \ldots, a_{(2^n-1)}, a_0)$ using $2^n - 1$ SWAP operations (Note that the Fredkin gate is a controlled-SWAP gate). This statement can be proved by mathematical induction:

(i) When n is 1, transforming (a_0, a_1) into (a_1, a_0) requires one SWAP operation.

(ii) Inductive hypothesis: When n is k, the statement is satisfied.

(iii) When n is $k+1$, the string is $(a_0, a_1, a_2, \ldots, a_{(2^{k+1}-1)})$. The string is divided into two parts: $(a_0, a_1, a_2, \ldots, a_{(2^k-1)})$ and $(a_{(2^k)}, a_{(2^k+1)}, a_{(2^k+2)}, \ldots, a_{(2^{(k+1)}-1)})$. By the inductive hypothesis, each string can be transformed into $(a_1, a_2, \ldots, a_{(2^k-1)}, a_0)$ and $(a_{(2^k+1)}, a_{(2^k+2)}, \ldots, a_{(2^{(k+1)}-1)}, a_{(2^k)})$ using $2^k - 1$ SWAP operations. Then, the total string becomes $(a_1, a_2, \ldots, a_{(2^k-1)}, a_0, a_{(2^k+1)}, a_{(2^k+2)}, \ldots, a_{(2^{(k+1)}-1)}, a_{(2^k)})$. Using another SWAP operation, this string can be transformed into $(a_1, a_2, \ldots, a_{(2^k-1)}, a_{(2^k)}, a_{(2^k+1)}, a_{(2^k+2)}, \ldots, a_{(2^{(k+1)}-1)}, a_0)$. Then, the total number of SWAP operations required is $2(2^k - 1) + 1 = 2^{(k+1)} - 1$.

□

Theorem A1. *Given n, synthesizing operator $C_{(2^k)}$ requires $2^n - 2^k$ Fredkin gates.*

Proof. It is sufficient to prove that a binary string $(a_0, a_1, a_2, \ldots, a_{(2^n-1)})$ can be transformed into $(a_{(2^k)}, a_{(2^k+1)}, a_{(2^k+2)}, \ldots, a_{(2^k-1)})$ using $2^n - 2^k$ SWAP operations, where additions and subtractions are modulo 2^n operations. This statement can also be proved by mathematical induction:

(i) When k is 0, the statement is true by Lemma 2.

(ii) Inductive hypothesis: When k is m, the statement is satisfied.

(iii) When k is $m+1$, the string is $(a_0, a_1, a_2, \ldots, a_{(2^n-1)})$. The string is divided into two parts: $(a_0, a_2, a_4, \ldots, a_{(2^n-2)})$ and $(a_1, a_3, a_5, \ldots, a_{(2^n-1)})$. According to the inductive hypothesis, each string can be transformed into the correct state using $2^n - 2^m$ SWAP operations. Then, the total number of SWAP operations required is $2(2^n - 2^m) = 2^{(n+1)} - 2^{(m+1)}$.

□

From Theorem 2, we prove that constructing a cyclic shift operator C requires $N \log_2 N - N + 1$ Fredkin gates.

Appendix B

In this appendix, we prove that the operator R_0 in the QSM algorithm can be an n-qubit operator, which is not explicitly presented in Ref. [4].

Theorem A2. *To build R_0 used in the QSM algorithm, it is not necessary to use an $(n+N+M)$-qubit controlled-Z gate; it is sufficient to use an n-qubit controlled-Z gate.*

Proof. Suppose that the second and third registers of $A^{-1}R_gA|k\rangle_n|0\rangle_N|0\rangle_M$ are $|0\rangle_N|0\rangle_M$ for an arbitrary n-qubit computational basis state $|k\rangle_n$. In this case, the states in the second and third registers in the QSM algorithm circuit before applying R_0 are always $|0\rangle_N|0\rangle_M$. This is because the operations for the QSM algorithm can be written as:

$$\begin{aligned} Q^rA|0\rangle_n|0\rangle_N|0\rangle_M &= (AR_0A^{-1}R_g)^rA|0\rangle_n|0\rangle_N|0\rangle_M \\ &= AR_0(A^{-1}R_gA)R_0(A^{-1}R_gA)R_0 \\ &\ldots(A^{-1}R_gA)R_0(A^{-1}R_gA)|0\rangle_n|0\rangle_N|0\rangle_M \end{aligned} \tag{A1}$$

Therefore, to prove Theorem 3, it is sufficient to prove that the states of the second and third registers in $A^{-1}R_gA|k\rangle_n|0\rangle_N|0\rangle_M$ are $|0\rangle_N|0\rangle_M$ for an arbitrary n-qubit computational basis state $|k\rangle_n$. The proof is as follows: Let a function $g:\{0,1,\ldots,N-1\}\to\{1,-1\}$ be defined by:

$$g(k)=\begin{cases}1, & |P_k\rangle_M=|0\rangle_M\\ -1, & |P_k\rangle_M\neq|0\rangle_M\end{cases} \tag{A2}$$

Then, the state $A^{-1}R_gA|k\rangle_n|0\rangle_N|0\rangle_M$ evolves as follows:

$$\begin{aligned} A^{-1}R_gA|k\rangle_n|0\rangle_N|0\rangle_M &= A^{-1}R_g\frac{1}{\sqrt{N}}\sum_{j=0}^{N-1}(-1)^{\vec{j}\cdot\vec{k}}|j\rangle_n|L_j\rangle_N|P_j\rangle_M \\ &= A^{-1}\frac{1}{\sqrt{N}}\sum_{j=0}^{N-1}(-1)^{\vec{j}\cdot\vec{k}+g(j)}|j\rangle_n|L_j\rangle_N|P_j\rangle_M \\ &= \frac{1}{N}\sum_{j=0}^{N-1}\sum_{l=0}^{N-1}(-1)^{\vec{j}\cdot\vec{k}+\vec{j}\cdot\vec{l}+g(j)}|j\rangle_n|0\rangle_N|0\rangle_M \end{aligned} \tag{A3}$$

□

Appendix C

In this paper, we introduced relative-phase encoding in the QSM algorithm to reduce circuit costs. This appendix explores how this strategy can be extended to quantum image processing, specifically within the framework of the NEQR model [31].

NEQR is a quantum image representation model that encodes both pixel positions and grayscale values in basis states. This contrasts with earlier models, such as the flexible representation of quantum images (FRQI) [32], where information is encoded in amplitude coefficients. NEQR enables precise image retrieval and efficient implementation of image processing tasks [31,33–35]. In this appendix, we investigate how relative-phase encoding can be integrated into NEQR to further optimize quantum image processing tasks, potentially enhancing computational efficiency and reducing circuit complexity.

A grayscale image of size $2^n\times 2^n$, with pixel values ranging from 0 to 2^q-1, is represented in NEQR as:

$$|I\rangle=\frac{1}{2^n}\sum_{Y=0}^{2^n-1}\sum_{X=0}^{2^n-1}|f(Y,X)\rangle_q|Y\rangle_n|X\rangle_n, \tag{A4}$$

where $|Y\rangle_n|X\rangle_n$ encodes pixel coordinates, and $|f(Y,X)\rangle_q$ represents grayscale values. If relative-phase encoding is introduced into the NEQR model, the image can be expressed as:

$$|I_r\rangle=\frac{1}{2^n}\sum_{Y=0}^{2^n-1}\sum_{X=0}^{2^n-1}z_{YX}\,|f(Y,X)\rangle_q|Y\rangle_n|X\rangle_n, \tag{A5}$$

where z_{YX} is an arbitrary complex number with a norm of 1. In the NEQR model, image retrieval is performed by measuring computational basis states [31]. Consequently, the NEQR images in Eqs. (A4) and (A5) yield the same probability distribution and can be considered equivalent representations of the same image.

Since the information in an NEQR image is encoded in basis states, fundamental image processing operations—such as intensity inversion and median filtering—modify pixel values and therefore act on computational basis states rather than amplitudes [31,33]. As a result, incorporating relative-phase information into basis states during image processing, which transforms the image from $|I\rangle$ to $|I_r\rangle$, does not alter the measurement outcome. This phase-encoding technique can contribute to reducing circuit costs.

For example, Figure A1a illustrates the construction of a one-qubit sorting module. The sorting module is a fundamental component utilized in various NEQR-based algorithms, including median filtering [33] and the NEQR-based QSM algorithm described in [30]. The design and detailed description of the sorting module can be found in Ref. [33,36]. In the circuit shown in Figure A1a, decomposing the Toffoli and four-qubit Fredkin gates using the method from Ref. [23] results in a T-count of 29. However, by employing the relative-phase gates, the T-count can be reduced to 16. Figure A1b presents the design of a four-qubit relative-phase Fredkin gate, which is constructed from a four-qubit relative-phase Toffoli gate [23] and two CNOT gates, following the same approach used in this study.

For the multi-qubit sorting module (see Figure A2), all multi-qubit Toffoli gates and four-qubit Fredkin gates can be replaced with their corresponding relative-phase gates. However, a general method for constructing multi-qubit relative-phase Toffoli gates has not yet been established. Therefore, developing an approach for constructing general multi-qubit relative-phase Toffoli gates remains an open research question and presents a promising direction for future studies.

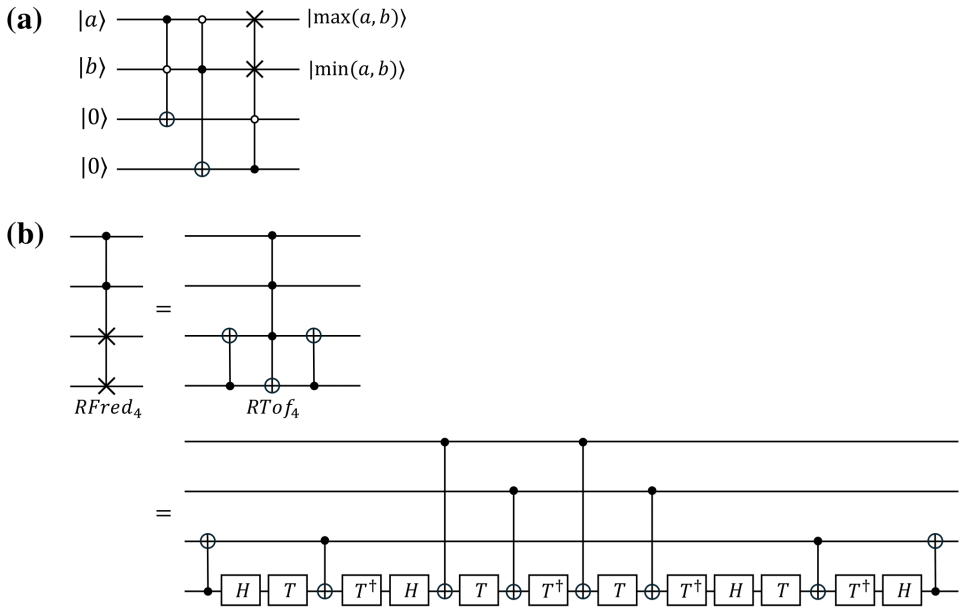

Figure A1. One-qubit sorting module. (**a**) Circuit diagram of the one-qubit sorting module employed in Ref. [33]. (**b**) Circuit of the four-qubit relative-phase Fredkin gate, constructed using the four-qubit relative-phase Toffoli gate proposed in Ref. [23], along with two CNOT gates. By replacing the Fredkin and Toffoli gates in Figure A1a with the four-qubit relative-phase Fredkin and relative-phase Toffoli gates from Ref. [23], the T-count of the one-qubit sorting module can be reduced from 29 to 16.

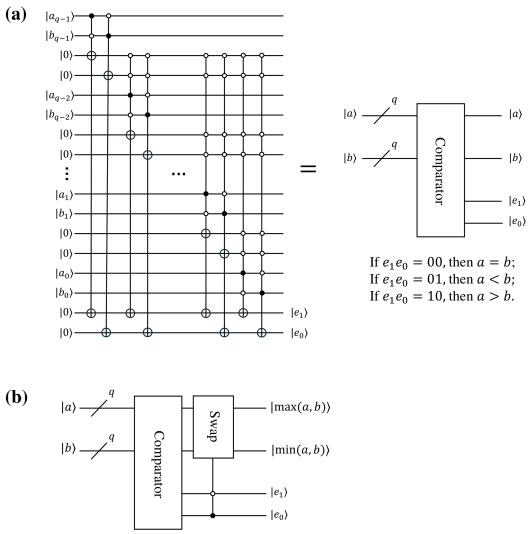

Figure A2. q-qubit sorting module. (a) Circuit diagram of the q-qubit comparator presented in Ref. [36], which compares the quantum states $|a\rangle = |a_{q-1}a_{q-2}\ldots a_0\rangle$ and $|b\rangle = |b_{q-1}b_{q-2}\ldots b_0\rangle$. (b) Circuit diagram of the q-qubit sorting module utilized in Ref. [33].

Appendix D

In this paper, we introduced the relative-phase Fredkin gate to reduce circuit costs in the QSM algorithm, drawing inspiration from the relative-phase Toffoli gate proposed in Ref. [23]. Given that relative-phase Toffoli gates have been successfully used in various quantum circuits to reduce circuit complexity, we anticipate that relative-phase Fredkin gates can also be applied to other circuits beyond the QSM algorithm. In this appendix, we present an example demonstrating how relative-phase Fredkin gates can be used to reduce circuit costs in a different quantum circuit context.

When a circuit D, which has a diagonal matrix representation, is positioned between two Fredkin gates, the standard Fredkin gates can be replaced with a relative-phase Fredkin gate and its Hermitian conjugate. This transformation is illustrated in Figure A3. According to Lemma 1 of this paper, any relative-phase Fredkin gate can be decomposed into the product of a diagonal matrix D_1 and a standard Fredkin gate:

$$\text{RFred} = D_1 \cdot \text{Fred}. \tag{A6}$$

Similarly, the Hermitian conjugate of the relative-phase Fredkin gate can be expressed as the product of a standard Fredkin gate and the Hermitian conjugate of D_1:

$$\text{RFred}^\dagger = \text{Fred} \cdot D_1^\dagger. \tag{A7}$$

By substituting these expressions into the overall circuit structure, the central diagonal matrix becomes $(D_1 \otimes I)^\dagger D(D_1 \otimes I)$, resulting in:

$$(D_1 \otimes I)^\dagger D(D_1 \otimes I) = D. \tag{A8}$$

Thus, the circuit structure remains unchanged under this replacement, completing the proof.

This method can be applied to the q-qubit comparator presented in Ref. [37] (see Figure A4) to reduce the overall circuit cost. The design and detailed explanation of the comparator are provided in Ref. [37]. When the q-qubit comparator is decomposed into the Clifford+T gate set using the Fredkin gate decomposition proposed in Ref. [4], the resulting T-count is $14q$. However, by replacing the Fredkin gates with the relative-phase Fredkin gates shown in Figure 1b and their Hermitian conjugates, the T-count can be reduced to $8q$.

The replacement procedure is as follows. As illustrated in Figure A5a, the CNOT gate highlighted by the red box in Figure A4 is decomposed into a controlled-Z gate and two Hadamard gates. This decomposition ensures that each subcircuit enclosed by the green boxes in Figure A5a—which lies between every pair of Fredkin gates—has a diagonal matrix representation. Once this condition is met, each pair of Fredkin gates can be replaced with a relative-phase Fredkin gate and its Hermitian conjugate (see Figure A5b).

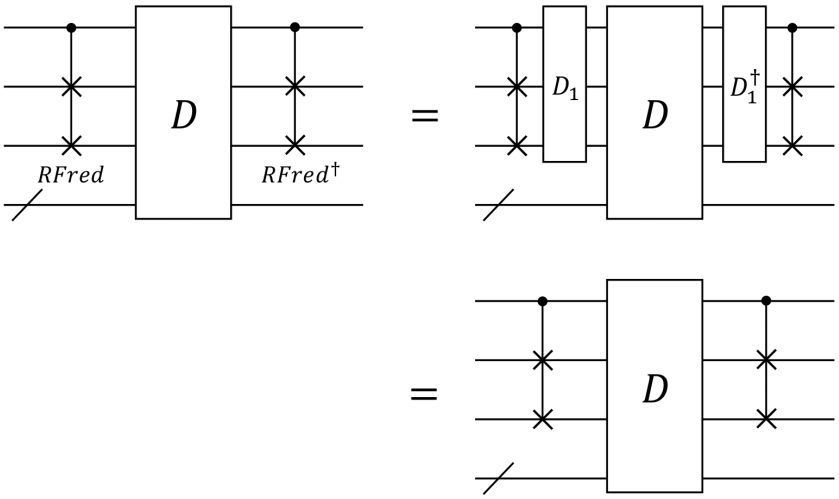

Figure A3. Equivalence transformation of the Fredkin–D–Fredkin structure, where D is a circuit block represented by a diagonal unitary matrix. A pair of Fredkin gates surrounding D can be replaced with a relative-phase Fredkin gate and its Hermitian conjugate. According to Lemma 1, each relative-phase Fredkin gate can be decomposed as $\text{RFred} = D_1 \cdot \text{Fred}$ and $\text{RFred}^\dagger = \text{Fred} \cdot D_1^\dagger$, where D_1 has a diagonal matrix representation. Substituting these into the circuit yields $(D_1 \otimes I)^\dagger D (D_1 \otimes I) = D$, showing that the transformation preserves the circuit's behavior.

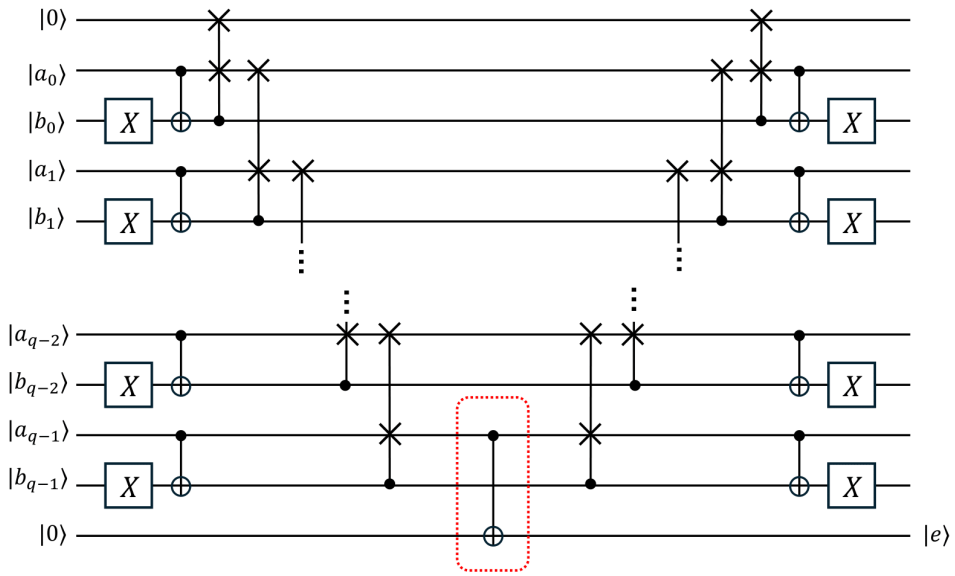

Figure A4. Circuit diagram of the q-qubit comparator presented in Ref. [37], which compares the quantum states $|a\rangle = |a_{q-1}a_{q-2}\ldots a_0\rangle$ and $|b\rangle = |b_{q-1}b_{q-2}\ldots b_0\rangle$. The output qubit $|e\rangle$ indicates the comparison result: if $|e\rangle = 0$, then $a \leq b$; if $|e\rangle = 1$, then $a > b$. When decomposed into the Clifford+T gate set using the Fredkin gate decomposition adopted in Ref. [4], the circuit requires a T-count of $14q$.

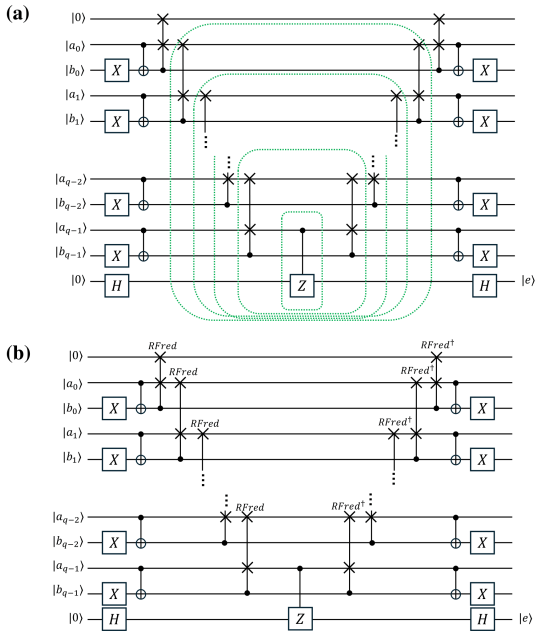

Figure A5. Optimization of the q-qubit comparator circuit presented in Ref. [37]. (**a**) Decomposition of the CNOT gate highlighted in the red box of Figure A4 into a controlled-Z gate and two Hadamard gates, which ensures that each subcircuit enclosed by the green boxes admits a diagonal matrix representation. (**b**) Optimized circuit obtained by replacing each pair of Fredkin gates in Figure A4 with the relative-phase Fredkin gate shown in Figure 1b and its Hermitian conjugate, thereby reducing the total T-count from $14q$ to $8q$.

References

1. L. K. Grover (1997). "Quantum mechanics helps in searching for a needle in a haystack." *Physical Review Letters,* 79, 325–328.
2. G. Brassard, P. Høyer, M. Mosca and A. Tapp (2002). "Quantum amplitude amplification and estimation", in *Contemporary Mathematics Series Millennium*, Vol. 305. New York: American Mathematical Society, pp. 53–74.
3. H. Ramesh and V. Vinay (2003). "String matching in ˜O($\sqrt{}$n + $\sqrt{}$m) quantum time." *Journal of Discrete Algorithms,* 1, 103–110.
4. P. Niroula and Y. Nam (2021). "A quantum algorithm for string matching." *npj Quantum Information*, 7, 37.
5. S. I. Hakak, A. Kamsin, P. Shivakumara, G. A. Gilkar, W. Z. Khan and M. Imran (2019). "Exact string matching algorithms: survey, issues, and future research directions." *IEEE Access*, 7, 69614–69637.
6. H. Schütze, C. D. Manning and P. Raghavan (2008). *Introduction to Information Retrieval*. Cambridge: Cambridge University Press.
7. R. R. Naik, M. B. Landge and C. N. Mahender (2015). "A review on plagiarism detection tools." *International Journal of Computer Applications,* 125.
8. M. K. Vijaymeena and K. Kavitha (2016). "A survey on similarity measures in text mining." *Machine Learning with Applications: International Journal*, 3, 19–28.
9. H. Lu, K. Zheng, B. Liu, X. Zhang and Y. Liu (2006). "A memory-efficient parallel string matching architecture for high-speed intrusion detection." *IEEE Journal on Selected Areas in Communications,* 24, 1793–1804.
10. R.-T. Liu, N.-F. Huang, C.-H. Chen and C.-N. Kao (2004). "A fast string-matching algorithm for network processor-based intrusion detection system." *ACM Transactions on Embedded Computing Systems*, 3, 614–633.
11. A. Lopez (2008). "Statistical machine translation." *ACM Computing Surveys*, 40, 1–49.
12. J. Tarhio and H. Peltola (1997). "String matching in the DNA alphabet." *Software: Practice and Experience,* 27, 851–861.
13. C. Ryu, T. Lecroq and K. Park (2020). "Fast string matching for DNA sequences." *Theoretical Computer Science*, 812, 137–148.
14. A. A. Karcioglu and H. Bulut (2021). "Improving hash-q exact string matching algorithm with perfect hashing for DNA sequences." *Computers in Biology and Medicine*, 131, 104292.

15. I. Alsmadi and M. Nuser (2012). “String matching evaluation methods for DNA comparison.” International Journal of Advanced Science and Technology, 47, 13–32.
16. E. T. Campbell, B. M. Terhal and C. Vuillot (2017). “Roads towards fault-tolerant universal quantum computatio.” *Nature*, 549, 172–179.
17. A. G. Fowler, M. Mariantoni, J. M. Martinis and A. N. Cleland (2012). “Surface codes: Towards practical large-scale quantum computation.” *Physical Review A*, 86, 032324.
18. E. Knill (2004). “Fault-tolerant postselected quantum computation: schemes.” arXiv:0402171.
19. S. Bravyi and A. Kitaev (2005). “Universal quantum computation with ideal Clifford gates and noisy ancillas.” *Physical Review A,* 71, 022316.
20. P. Aliferis, D. Gottesman and J. Preskill (2006). “Quantum accuracy threshold for concatenated distance-3 codes.” *Quantum Information & Computation,* 6, 97–165.
21. A. G. Fowler, A. M. Stephens and P. Groszkowski (2009). “High-threshold universal quantum computation on the surface code.” *Physical Review A*, 80, 052312.
22. A. Montanaro (2017). “Quantum pattern matching fast on average.” *Algorithmica*, 77, 16–39.
23. D. Maslov (2016). “Advantages of using relative-phase Toffoli gates with an application to multiple control Toffoli optimization.” *Physical Review A,* 93, 022311.
24. J. A. Smolin and D. P. DiVincenzo (1996). “Five two-bit quantum gates are sufficient to implement the quantum Fredkin gate.” *Physical Review A,* 53, 2855–2856.
25. A. Javadi-Abhari, M. Treinish, K. Krsulich, C.J. Wood, J. Lishman, J. Gacon, S. Martiel, P.D. Nation, L.S. Bishop, A.W. Cross, B.R. Johnson and J.M. Gambetta (2024). “Quantum computing with Qiskit.” arXiv:2405.08810.
26. P. Selinger (2013). “Quantum circuits of T-depth one.” *Physical Review A*, 87, 042302.
27. Y. Nam, Y. Su and D. Maslov (2020). “Approximate quantum Fourier transform with O(n log(n)) T gates.” *npj Quantum Information*, 6, 26.
28. K. Oonishi, T. Tanaka, S. Uno, T. Satoh, R. Van Meter and N. Kunihiro (2021). “Efficient construction of a control modular adder on a carry-lookahead adder using relative-phase Toffoli gates.” *IEEE Transactions on Quantum Engineering,* 3, 1–18.
29. C. Jones (2013). “Low-overhead constructions for the fault-tolerant Toffoli gate.” *Physical Review A,* 87, 022328.
30. K. Prousalis, A. Kydros and N. Konofaos (2025). “A quantum string-matching algorithm.” *Advanced Quantum Technologies*, 8, 2400250.
31. Y. Zhang, K. Lu, Y. Gao and M. Wang (2013). “NEQR: A novel enhanced quantum representation of digital images.” *Quantum Information Processing*, 12, 2833.
32. P. Q. Le, F. Dong and K. Hirota (2011). “A flexible representation of quantum images for polynomial preparation, image compression, and processing operations.” *Quantum Information Processing***,** 10, 63.
33. S. Jiang, R.-G. Zhou, W. Hu and Y. Li (2019). “Improved quantum image median filtering in the spatial domain.” *International Journal of Theoretical Physics,* 58, 2115.
34. P. Li, X. Liu and H. Xiao (2018). “Quantum image median filtering in the spatial domain.” *Quantum Information Processing,* 17, 1.
35. T. Li, P. Zhao, Y. Zhou and Y. Zhang (2023). “Quantum image processing algorithm using line detection mask based on NEQR.” *Entropy,* 25, 738.
36. D. Wang, Z.-H. Liu, W.-N. Zhu and S.-Z. Li (2012). “Design of quantum comparator based on extended general Toffoli gates with multiple targets.” *Computer Science***,** 39, 302.
37. H.-Y. Xia, H. Zhang, S.-X. Song, H. Li, Y.-J. Zhou and X. Chen (2020). “Design and simulation of quantum image binarization using quantum comparator.” *Modern Physics Letters* A 35, 2050049.