Quantum Information Processing (2023) 22:381
https://doi.org/10.1007/s11128-023-04142-7

MPMCT gate decomposition method reducing T-depth quickly in proportion to the number of work qubits

Jongheon Lee[1] · Yousung Kang[1] · You-Seok Lee[1] · Boheung Chung[1] · Dooho Choi[2]

Received: 9 July 2023 / Accepted: 2 October 2023

Abstract

We propose a method for efficient mixed polarity multiple controlled Toffoli (MPMCT) gate decomposition from the perspective of a cost metric related to Toffoli gates, namely Toffoli-depth. When using the technique presented in a previous study, there is a range in which Toffoli-depth (consequently T-depth) of the implemented circuit increases proportionally as the number of provided (clean) work qubits increases. In other words, using the previous technique may result in more inefficient MPMCT gates even though the number of helpful work qubits has increased. In this work, a technique is devised to provide sufficient help from clean work qubits at the central part of the implemented circuit as many as possible, thereby addressing the issues with the previous technique. Meanwhile, one of the representative algorithms that use MPMCT gates is Grover's algorithm. We show the implementation results for MPMCT gates according to the number of work qubits, using Grover's algorithm as an example. It is experimentally demonstrated that T-depth decreases much more quickly when using our method than the previous method.

✉ Dooho Choi
doohochoi@korea.ac.kr

Jongheon Lee
jonghun0805@etri.re.kr

Yousung Kang
youskang@etri.re.kr

You-Seok Lee
yslee75@etri.re.kr

Boheung Chung
bhjung@etri.re.kr

[1] Cryptography and Authentication Base Technology Research Section, Electronics and Telecommunications Research Institute, 218, Gajeong-ro, Yuseong-gu, Daejeon 34129, Korea

[2] Department of AI Cyber Security, College of Science and Technology, Korea University Sejong, 2511, Sejong-ro, Jochiwon-eup, Sejong 10587, Korea

Published online: 17 October 2023

Keywords Quantum circuit · MPMCT gate · T-depth · Toffoli-depth

1 Introduction

Operations in quantum computing are typically represented using quantum (reversible) circuits, and therefore, various studies have been conducted on reversible circuit synthesis and circuit optimization [1–6]. The design efficiency of a circuit determines the design cost and latency for the desired reversible operation. When making circuits, among various gate libraries, the Clifford+T gate library is commonly used when considering FTQC (fault-tolerant quantum computation) in particular [7]. It is known that a T (or $T^\dagger$) gate, one of the non-Clifford gates, in this standard universal fault-tolerant gate library is more costly to design and take longer to execute than Clifford gates. [8]. In other words, T gates are dominant factors in the FTQC setting in terms of the running time and the implementation cost. T-depth (depth formed by T gates that operate non-parallelly) may determine the execution time of the circuit, and T-count (the number of T gates) may decide the design cost and the required fidelity of the gates. That is, T-depth and T-count, which are cost metrics associated with T gates, are important considerations. As an example, the quantum resources required for magic state distillation in FTQC setting are determined by these two cost metrics. T and $T^\dagger$ gates are primarily employed within Toffoli gate, a widely recognized composite gate. As a result, cost metrics about Toffoli gates can be utilized instead of those associated with T gates such as Toffoli-depth and Toffoli-count [9].

Meanwhile, several studies focused on specific gates such as MPMCT (mixed polarity multiple controlled Toffoli) gates and how to decompose these gates to Toffoli gates [1, 2, 6, 10]. In this work, we present an advanced method to reduce Toffoli-depth of a given MPMCT gate and thereby dependently reduce T-depth. Of course, the resulting circuit's T-depth can be reduced further by previous works [3–5]. One of the existing MPMCT gate decomposition methods was announced [10], but its cost metrics are phase-depth & phase-count, which are the generalizations of T-depth & T-count in terms of phase. In their work, the gates may deal with phases smaller than T gate's phase, so it may be expected that more complicated work is required when designing a real quantum computer. Other MPMCT gate techniques use T-depth & T-count as cost metrics [1, 2, 6]. Another previous method divided the number of CWQs (clean work qubits) into three categories based on the number of MPMCT gate control parts [2]. However, when this method is applied, the resulting circuit's T-depth (or T-count) sometimes increases as the number of CWQs or DBQs (dirty borrowed qubits) used for implementing the given MPMCT gate increases. Descriptions for the terminologies of work qubits (or ancilla qubits) are shown in the next section. They just showed results using a graph when the number of controls is too small. In other words, using this method may result in a more inefficient circuit despite adding more quantum resources. This problem arises because the central part of the resulting circuit is not sufficiently supported by CWQs. To address this issue, we present an advanced method that employs CWQs properly at all circuit regions. This method ensures that the number of work qubits and T-depth do not increase simultaneously.

The proposed technique is demonstrated using Grover's algorithm with the SHA-256 and SHA3-256 hash algorithms as examples in Sect. 4 [3, 11]. In 1996, Grover developed an algorithm that uses quantum computing to retrieve a pre-image or key for a given cryptosystem much more quickly than any classical algorithm [12]. Grover's algorithm consists of Oracle operators and Diffusion operators, both of which use MPMCT gates. However, the implementation way of MPMCT gate in each operator should be different because the state of the work qubits used to help implement each gate may differ within each operator. Depending on the cryptosystem quantum circuit used, the states of the work qubits that can help implement the MPMCT gate are different. We will show the quantum resources for MPMCT gates that can be used in Grover's algorithm according to the number of CWQs and DBQs. We will also show the quantum resources required to implement the attack algorithm circuit for each cryptosystem. In the case of SHA-256, it can be seen that the quantum resources are less required than those presented in a previous study [9]. Of course, the proposed MPMCT gate technique is not limited to Grover's algorithm and can be used in other quantum algorithms, such as quantum random-walk-based algorithms, indicating its high usability [13–15].

The rest of this paper is organized as follows. Section 2 begins by introducing terminologies for kinds of qubits according to the purpose of use and the states in the quantum circuit. We briefly describe previous lemmas and techniques for MPMCT gates decomposition and discuss the limitations of these. In addition, we mention Grover's algorithm, which is used as a specific example in our proposed method, and which MPMCT gates are needed in this algorithm. In Sect. 3, we present our advanced technique, which is divided into four categories, and explain implementation ways for each category. We note that, like previous studies, we utilize well-known lemmas to decompose MPMCT gates into Toffoli gates as mentioned in Sect. 2. In Sect. 4, we demonstrate the superiority of the proposed technique by implementing Grover's algorithm. Specifically, we try to implement Grover's algorithm for SHA-256 or SHA3-256 cryptosystems and compare the quantum resources required for security strength measurement with those in previous studies. In the last section, we summarize and point out the limitations of this study and suggest future research directions.

2 Background and previous work

In this section, the types of qubits, an MPMCT gate, and the techniques presented in previous studies are mentioned. Grover's algorithm, which will be used as an application example of our proposed technique, is also briefly mentioned.

2.1 Qubit and MPMCT gate

A qubit is a fundamental unit of quantum information and can be represented as a two-dimensional vector that belongs to the (projective) Hilbert space $\mathcal{H}$. In contrast to classical bits, a qubit can exist in states with various phases and values due to its ability to be expressed as a linear combination of $|0\rangle$ and $|1\rangle$. Quantum circuits represent reversible operations, which are unitary in complex space. As a result, the size of the state of qubits remains constant throughout the circuit. Depending on their states and usage purposes in the circuit, qubits are referred to by different names. We will utilize the following various types of qubits [9]:

1. Data qubits are the qubits that hold the input data in a quantum algorithm. For example, in the case of SHA-256, the data qubits represent the pre-image search space. If the length of the plaintext (or pre-image) is m-bit, the quantum circuit requires m data qubits. Therefore, in Grover's algorithm, the data qubits represent the search space for the solution. All qubits that are not data qubits are work qubits or ancilla qubits, which assist in performing specific operations in the algorithm.
2. Clean work qubits (CWQs) are work qubits whose initial states are known and can assist in performing certain operations in quantum circuits. CWQs can be initialized through an uncomputation (restoration or clearing) step after a specific operation is finished, which restores them to their original state. The restored CWQs can then be used in subsequent operations with the same initial state. Typically, CWQs are initialized to the $|0\rangle$ states.
3. Dirty borrowed qubits (DBQs) are work qubits whose initial states are unknown or entangled, so they are considered to be in arbitrary states before being provided in a specific operation. Unlike CWQs, these work qubits are less effective due to these arbitrary initial states. When a sub-circuit representing the specific operation is made, the restoration step for DBQs' states may be included in the case of utilizing their original states in the following sub-circuit, which makes the circuit design more complicated than the case for CWQs.

Quantum circuits are commonly visualized as two-dimensional diagrams, with data qubits placed on the top and work qubits on the bottom. In the data qubits range, the least significant bit (LSB) data qubit is typically located at the top, while the most significant bit (MSB) is placed at the bottom. That is, data qubits are sequentially arranged from top to bottom according to the size of the binary digit. The flow of time in a quantum circuit is from left to right.

Quantum gates used in quantum circuits correspond to specific unitary operations. In this context, the Clifford+T gates are typically consdiered as elementary gates, as mentioned in equation (1). One commonly used composite gate is Toffoli (or C^2NOT, a doubly controlled-NOT) gate, which has T-depth 3 and T-count 7 [16] (Fig. 1). (This T-depth value is optimal with no work qubits and no limitation for CNOT-count.) Toffoli gate is the quantum analogue of the AND gate in classical circuits ($Toffoli : |x_1x_2x_3\rangle \rightarrow |x_1x_2((x_1 \wedge x_2) \oplus x_3)\rangle$). Toffoli-count and Toffoli-depth are cost metrics associated with Toffoli gate, which frequently employs T gates. Therefore, these metrics can be used instead of T-count and T-depth, which are considered significant when considering FTQC. Toffoli-count is the number of Toffoli gates in a quantum circuit, and Toffoli-depth is the number of non-parallel processing of Toffoli gates in a quantum circuit. Our method tries to express a given MPMCT gate as a circuit with optimized Toffoli-depth. After applying the our presented method, the resulting circuit's T-depth can be reduced using previous T-depth reduction techniques [3–5, 9]. That is, as in these previous studies [2, 6], FTQC computing which deals with T-depth (or T-count) is considered more rather than NISQ (noisy intermediate-

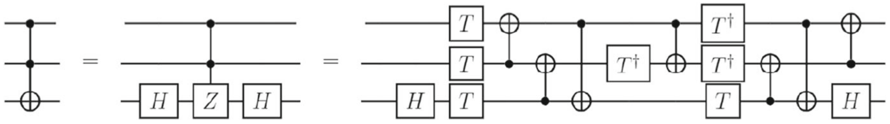

Fig. 1 A Toffoli gate [16]. Toffoli-depth, which is a metric for Toffoli gates, are used when presenting our MPMCT gate decomposition method

scale quantum) computing which deals with CNOT-count (or CNOT-depth) [17–19]. In NISQ computing, various qubit placements (or the qubit connectivities) such as linear and square-grid are considered, while in this study, the circuit is designed when qubit connectivity is assumed to be all-to-all, that is, at the logical level. Therefore, swapping operations for adjacent qubit placement are not considered [19]. In one study [18], various logical equivalent circuits were created in terms of CNOT-count, while this work makes circuits with various T-depths. Then, the one with the smallest T-depth value is selected.

$$\begin{aligned} T &: |x_1\rangle \rightarrow e^{\frac{\pi i}{4}x_1}|x_1\rangle & H &: |x_1\rangle \rightarrow \frac{|0\rangle + (-1)^{x_1}|1\rangle}{\sqrt{2}} \\ P &: |x_1\rangle \rightarrow e^{\frac{\pi i}{2}x_1}|x_1\rangle & X &: |x_1\rangle \rightarrow |x_1 \oplus 1\rangle \\ Z &: |x_1\rangle \rightarrow (-1)^{x_1}|x_1\rangle & CNOT &: |x_1x_2\rangle \rightarrow |x_1(x_1 \oplus x_2)\rangle \end{aligned} \tag{1}$$

An MPMCT gate T(C$_1$,C$_2$,t), which is dealt with in this paper, can be defined as follows [2].

Definition 1 Given a set X = {x$_1$, $\cdots$, x$_n$}of lines in a quantum circuit, an MPMCT gate T(C$_1$,C$_2$,t) (or T(C,t)) consists of three line sets C$_1$,C$_2$,and {t}. (The set of all control lines is represented by C (= C$_1 \bigcup$ C$_2$).)

- the set of on-control lines C$_1$={x$_{11}$, $\cdots$, x$_{1i}$} $\in$ X,
- the set of off-control lines C$_2$={x$_{21}$, $\cdots$, x$_{2j}$} $\in$ X,
- and a target line t $\in$ X/(C$_1 \bigcup$ C$_2$)

The target line undergoes a bit-flip (or inversion) if all the states of the on-control lines are true ($|1\rangle$) and all the states of the off-control lines are false ($|0\rangle$). This means that when passing through the MPMCT gate, the states of the qubits on the control lines remain unchanged, while the state of the qubit corresponding to the target line $|t\rangle$ is mapped to $|t \oplus (x_{11} \wedge \cdots \wedge x_{1i} \wedge \overline{x_{21}} \wedge \cdots \wedge \overline{x_{2j}})\rangle$ if conditions hold.

The MPMCT gate can be constructed by adding NOT gates or CNOT gates to a non-MPMCT gate, as shown in Fig. 2. This figure shows the simplest version, MPT (mixed polarity Toffoli) gate. The MPMCT gate is also called the MCT gate or the MP-C^{c}NOT (mixed polarity c-controlled-NOT) gate.

2.2 Previous works for decomposing MPMCT gates

Previous studies have proposed methods to decompose MCT gates into multiple Toffoli gates using different numbers of CWQs and DBQs. First, the lemmas presented in

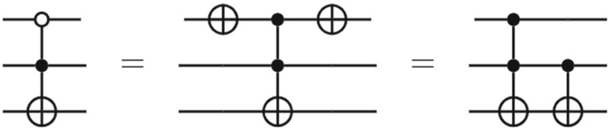

Fig. 2 An MPT (A mixed polarity Toffoli) gate. For this gate to work, the state of the qubit in the first control line should be false (|0⟩) and the state of the qubit in the second control line should be true (|1⟩)

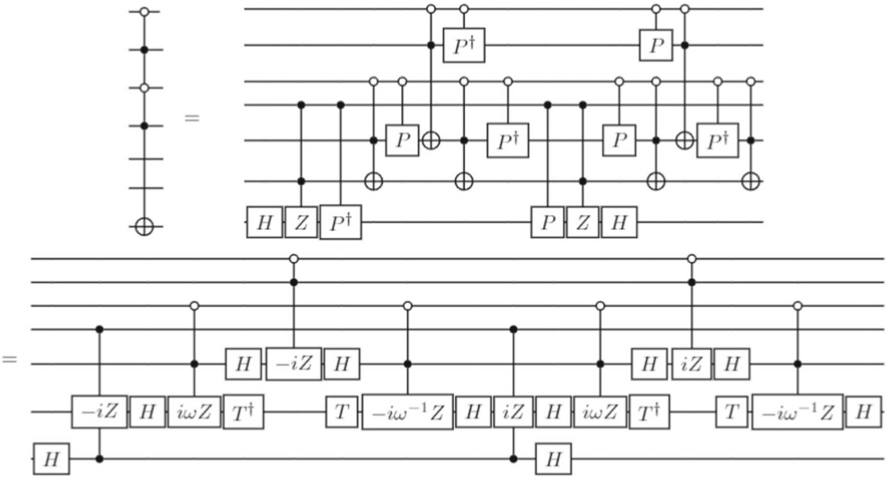

Fig. 3 An MP-C^4NOT (a mixed polarity 4-controlled-NOT) gate. By using two DBQs, the gate can be represented with a circuit with Toffoli-depth (Toffoli-count) of 8 and T-depth of 12. [3–5]

different studies are briefly mentioned. Then, the algorithm in one previous study is mentioned in detail [2]. And then another study is briefly mentioned [6].

2.2.1 Lemmas for decomposing MPMCT gates

These Lemmas can be utilized to decompose the MPMCT gate into several Toffoli gates, depending on the type and number of given work qubits.

Lemma 1 *For c≧3, a C^cNOT gate can be represented as a gate with Toffoli-depth (& Toffoli-count) of 4(c-2) and T-depth of 4(c-1) given c-2 DBQs* [1–3, 20].

Lemma 1 is a result of combining MPMCT decomposition methods and T-depth (T-count) reduction techniques [3–5]. By utilizing this lemma, a circuit with a T-count of 12c-20 can be implemented, where Width (the total number of qubits) should be at least 2c-1. To provide an example for this Lemma 1, Fig. 3 illustrates the MP-C^4NOT gate. It can be observed that the MP-C^4NOT gate can be represented by a circuit with Toffoli-depth (& Toffoli-count) of 8 and T-depth of 12.

Lemma 2 *For c≧4 and a single DBQ, a C^cNOT gate can be implemented with T-depth of 8c-20* [1, 2, 21]. *This implementation requires not only Toffoli gates but two controlled-V & controlled-$V^\dagger$ gate pairs, where V gate is defined by a gate satisfying*

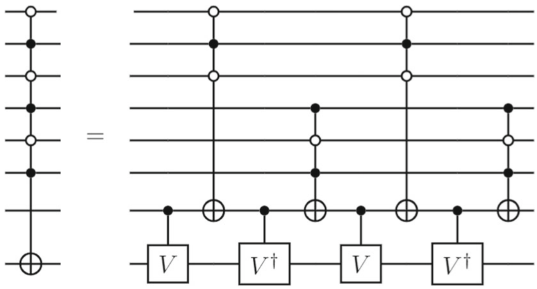

Fig. 4 An MP-C^{6}NOT gate implementation by MI (Miller) mapping [1, 21]. Four 3-controlled sub-MCT gates can be decomposed into 16 Toffoli gates, resulting in a total T-depth of 28

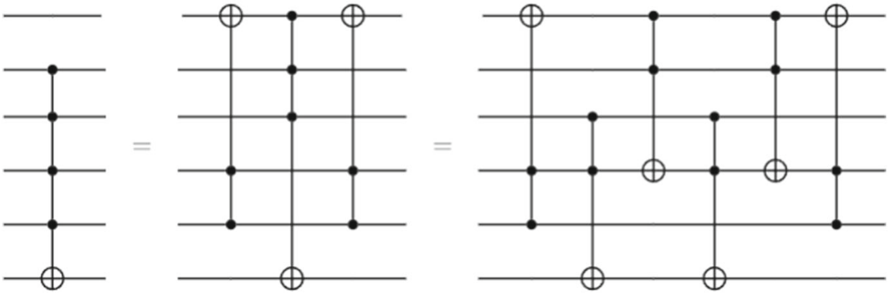

Fig. 5 A C^{4}NOT gate with 1 CWQ. A C^{4}NOT gate can be expressed using a circuit that has Toffoli-depth of 6 and T-depth of 11

$V^2 = X$. For c=4, Toffoli-depth (Toffoli-count) is 4, and for c=5, it is 12. When c≧6, the MCT gate can be implemented with Toffoli-depth of 8c-32.

Lemma 2 is referred to as MI (or Miller) mapping, which utilizes controlled-V & controlled-V† gate pairs. Since the V gate is $\sqrt{X}$, $V : |x\rangle \rightarrow \frac{(1+(-1)^x i)|0\rangle+(1-(-1)^x i)|1\rangle}{2}$, where $x \in \{0, 1\}$ [16]. An example of MI mapping with six controls is depicted in Fig. 4, where the MP-C^{6}NOT gate is composed of four C^{3}NOT gates and two controlled-V/V† gate pairs. This configuration results in a C^{6}NOT gate with T-depth of 28.

Lemma 3 *For c≧4 and a single CWQ, a C^{c}NOT gate can be expressed as a circuit with T-depth of 6(c-2) when c is even and 6(c-2)-2 when c is odd [1, 2]. When c=4, Toffoli-depth (Toffoli-count) is 6, and when c=5, it is 10. When c≧6, Toffoli-depth is 6c-20 when c is even and 6c-22 when c is odd.*

This Lemma 3 describes the NC mapping in previous studies [1, 22]. One previous study stated that T-depth is 6(c-2) when c is even and T-depth is 6(c-2)+2 when c is odd [1]. However, according to the logic in their study, T-depth for odd c could be 6(c-2)-2 instead of 6(c-2)+2.

As an example, we have provided circuit diagrams for the C^{4}NOT and C^{9}NOT gates in Figs. 5 and 6, respectively. These circuits' T-depth values result in T-depths of 12 and 40, respectively.

In Fig. 5, it can be observed that T-depth of the C^{4}NOT gate can be reduced up to 11 instead of 12. First, the C^{4}NOT gate is decomposed into 6 Toffoli gates, which are

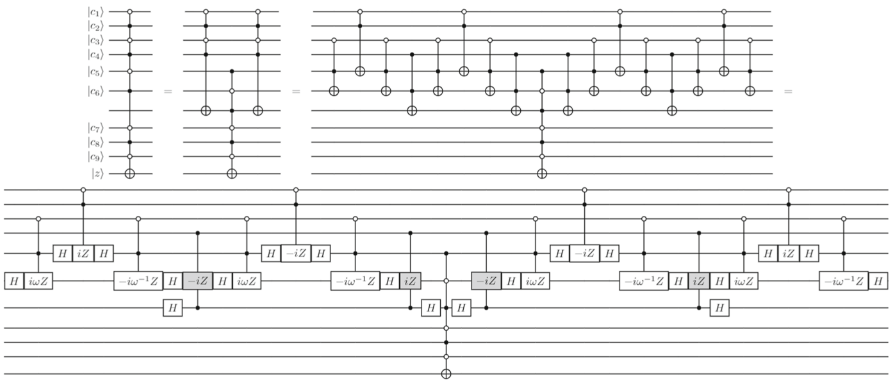

Fig. 6 A C^9NOT gate decomposition with 1 CWQ. Appropriate pairing for the C^2(iZ) & C^2(-iZ) gates painted in black with each other can reduce T-depth (or T-count) by 4, according to previous studies [1, 3]. Based on the figure, the total T-depth is 40. However, if the C^6NOT gate in the central part is decomposed properly, T-depth may be shared between the C^4NOT gates and the C^6NOT gate, resulting in further T-depth reduction. Therefore, it may be possible that the total T-depth is less than 40

then converted into 6 C^2(iZ) or C^2(-iZ) gates using T-depth reduction methods. At this point, the leftmost two C^2(iZ) & C^2(-iZ) gates can share one T-depth, resulting in an additional reduction of T-depth by one. Thus, T-depth of the C^4NOT gate can become 11.

For the C^9NOT gate, it is decomposed into 2 C^4NOT gates and 1 C^6NOT gate, and then further decomposed into 32 Toffoli gates. T-depth can be reduced from 44 to 40 by pairing the C^2(iZ)/C^2(-iZ) gates that are painted in black in the circuit diagram. (That is, further T-depth reduction can be achieved by pairing the first and fourth gates, and the second and third gates, as described in [3].) This figure shows that by decomposing the C^6NOT gate appropriately, T-depth can be shared between the C^4NOT gates and the C^6NOT gate, potentially leading to a reduction in T-depth beyond 6(c-2) or 6(c-2)-2 for c≧4. In other words, Toffoli gates forming the sub-MCT gate in the center and Toffoli gates composing the front & back sub-MCT gates may be designed to share the time slice (stage).

Given these observations, some readers may wonder to what extent T-depth can be reduced related to this Lemma. In next section, Lemma 5 provides a lower bound on T-depth of the C^cNOT gate when there is one CWQ but this lemma also considers the number of DBQs in the circuit. If there are enough DBQs in addition to one CWQ, a more efficient design for the C^cNOT gate can be achieved (Lemma 5).

Now let us consider the scenario where there are sufficiently many CWQs. If there are CWQs enough, that is, if the number of CWQs is close to the number of controls of a given C^cNOT gate, then this gate can be implemented as a circuit with Toffoli-depth $\mathcal{O}$(log c) (Lemma 4).

Lemma 4 *For c≧3, a C^cNOT gate can be expressed as a circuit with Toffoli-depth $2\lceil log_2 c\rceil$-1 and T-depth $2\lceil log_2 c\rceil$+2 when there are c-2 CWQs* [2, 5]. *If there are c-1 CWQs, the gate can be expressed with T-depth $2\lceil log_2 c\rceil$+2 when c is even, or T-depth $2\lceil log_2 c\rceil$ when c is odd. If the number of CWQs is c, the C^cNOT gate can be*

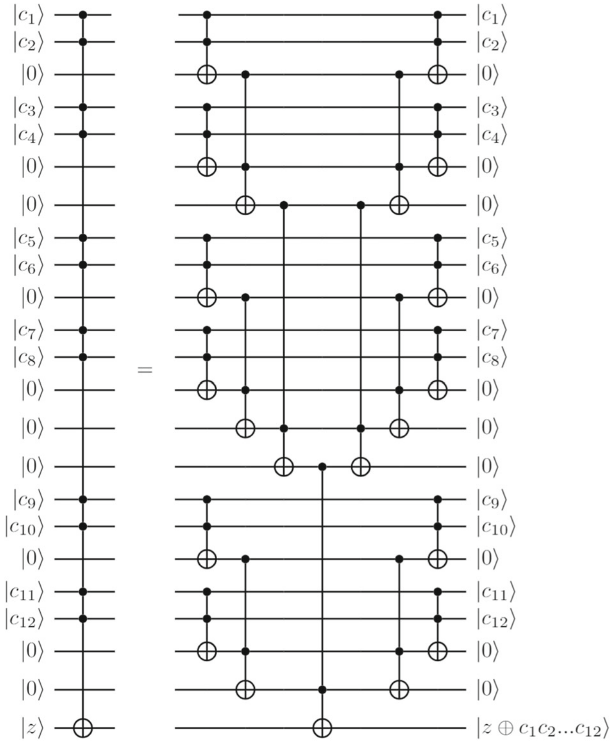

Fig. 7 A C^{12}NOT gate with 10 CWQs. This gate can be implemented using Toffoli-depth 7 and T-depth 10 gates. This is achieved by reducing the number of controls to be handled by about half each time passing through a time slice formed by Toffoli gates

implemented with T-depth 2$\lceil log_2 c\rceil$. If the number of CWQs is equal to or more than c+1, it can be implemented with T-depth 2$\lceil log_2 c\rceil$-1. Therefore, when the number of CWQs is c+1 or greater, T-depth of the C^cNOT gate can be reduced to the same value as its Toffoli-depth.

Furthermore, when the number of CWQs for a C^cNOT gate is c-2, the gate can be implemented with Toffoli-count 2c-3 and T-count 8c-9. A proof of this lemma is provided in “Appendix A.” Fig. 7 provides an example implementation of the C^{12}NOT gate, which has Toffoli-depth 7, and Toffoli-count 21, and utilizes a total of 10 CWQs.

The lemmas discussed are summarized in Table 1, which reveals that when the number of available CWQs is sufficient, Toffoli-depth (T-depth) can be expressed in a logarithmic form, not linear form. This means that in order to design an efficient MPMCT gate, it is important to manage the number of available CWQs effectively, so that Lemma 4, which enables circuit designs with Toffoli-depth $\mathcal{O}$(log n), can be utilized. In our proposed method, one of these lemmas is selected and used in the central step. If there are no CWQs at all and there are not many DBQs, the DBQs-dedicated algorithm from the previous study is used [6]. As a side note, when decomposing an MPMCT gate for c≧3 using a quantum circuit based on the Clifford+T (or NCT,

Table 1 Lemmas for C^{c}NOT gate decomposition (c≧3)

	#Control	#CWQ	#DBQ	T-depth	Toffoli-depth	Remark
Lemma 1 [1–3, 20]	c≧3	0	c-2	4c-4	4c-8	–
Lemma 2 [1, 2]	c≧4	0	1	8c-20	8c-32 for c≧6	CV/CV† pairs are used
Lemma 3 [1–3]	c≧4	1	0	6c-12(-2)	6c-20(-2) for c≧6	–
Lemma 4 [2, 5, 16]	c≧3	c-2	0	2$\lceil log_2 c\rceil$+2	2$\lceil log_2 c\rceil$-1	–
Lemma 5	c≧4	1	c-5	4c-5	4c-10	-

It is clear from the table that the efficient design of MPMCT gates requires effective management with CWQs available, so Lemma 4 can be applied to achieve Toffoli-depth $\mathcal{O}(\log n)$. Meanwhile, Lemma 5 provides a lower bound on T-depth of the C^{c}NOT gate compared to Lemma 3 when using 1 CWQ

which is NOT, CNOT, and Toffoli) gate library, it is known that it may be necessary to utilize at least one work qubit [23]. However, work qubits may not be required when decomposing MCT gates if elementary gates that can handle smaller (global) phases are employed, as suggested in [10, 24].

2.2.2 Previous methods for decomposing MPMCT gates

The lemmas discussed previously only considered cases where the number of available CWQs was either 0, 1, or close to the number of control lines of a given MPMCT gate. There have been previous studies on the decomposition of MPMCT gates when given various numbers of CWQs and DBQs [2, 6]. One study [2] categorizes the number of CWQs into three ranges based on the number of control lines. The number of controls for a C^{c}NOT gate is denoted as c, the number of CWQs as k, and the number of DBQs as d. The ith CWQ is denoted as a_i, where i ranges from 1 to k.

The first case is where k is more than or equal to 1 and less than or equal to approximately $c/2$. In this range, the C^{c}NOT gate can be decomposed as follows.

1. At the front step, the control lines of the C^{c}NOT gate T(C,t) are divided into $k+1$ groups $C_1, \cdots, C_{k+1}$. The first k groups correspond to the control parts of sub-MCT gates. The target parts consist of the CWQs (a_i). These k sub-MCT gates T(C_i, a_i) are placed in the front part of the quantum circuit to be executed at the same time. These gates do not share any control lines or target lines. As further explained below, this step may consist of several stages (time slices).
2. At the central step, we install a sub-MCT gate T($C_{k+1} \cup \{a_1, \cdots, a_k\}$,t) that takes control lines previously used as target parts in the front step. This is done using an original target line t and including control lines in the last group C_{k+1}.
3. At the back step, we place the same k MCT gates created in the front step to initialize the values of the CWQs as an uncomputation step. If the front step consists of multiple stages, the back step is organized in the reverse order of the design of the front step.

In this decomposition method, the gate efficiency (Toffoli-depth or T-depth) is determined depending on how the control lines are classified into $C_1, \cdots, C_{k+1}$. The sum of their sizes is equal to the total number of control lines c (= $|C_1| + \cdots + |C_k| +$

$|C_{k+1}|$). The paper provides the condition expression for this grouping (2). The reason for this conditional expression is to secure the number of DBQs so that Lemma 1 can be used in every step.

$$\frac{c-2k-d-1}{2} \leq |C_{k+1}| \leq \frac{c+2-k+d}{2} \tag{2}$$

They suggest a way to determine the sizes of the control groups $C_1, \cdots, C_{k+1}$ for the above decomposition process.

1. First, $|C_{k+1}|$ is set to the maximum possible value, considering equation (2). We distribute the remaining $n - |C_{k+1}|$ control lines in $C_1, \cdots, C_k$ as evenly as possible, with a maximum difference of 1 between each group.
2. We move the control lines from C_{k+1} to $C_1, \cdots, C_k$ while the existing maximum value of $|C_1|, \cdots, |C_k|$ maintains. Equation (2) should still be satisfied.
3. If k is more than 2 and there are more than 3 control lines that can be moved from C_{k+1} to $C_1, \cdots, C_k$, then 2 control lines are moved, one to C_1 and the other to C_2, while still satisfying equation (2). After completing this step 3, the algorithm returns to step 2 and repeats it with the updated sets $C_1, \cdots, C_{k+1}$. If the number of control lines that can be moved from C_{k+1} to $C_1, \cdots, C_k$ is not more than 3, this algorithm terminates.

In step 3, when going back to step 2, the control lines from C_{k+1} may be further distributed to $C_3, \cdots, C_k$ since the maximum value among $|C_1|, \cdots, |C_k|$ has increased by 1. A detailed explanation of this can be found in a previous paper [2].

The second case is when k is more than or equal to approximately $c/2$ and less than $c-2$. In this case, $\lfloor c/2 \rfloor$ CWQs are used for the first time slice, which is composed of $\lfloor c/2 \rfloor$ Toffoli gates that operate simultaneously.

In the second time slice, which is within the front step, the remaining $c - \lfloor c/2 \rfloor$ control lines should be processed using the remaining $k - \lfloor c/2 \rfloor$ CWQs. The number of DBQs available at this stage is increased by $2\lfloor c/2 \rfloor$. From this stage, one of the above two strategies is selected and applied recursively. This means that if the number of remaining CWQs is more than half of the number of remaining controls, time slices are constructed only with Toffoli gates, as mentioned earlier. On the other hand, if the number of remaining CWQs is insufficient, the strategy mentioned in the first case is followed. For example, if $k = c - 3$, the front step will consist of several time slices consisting only of Toffoli gates, and consequently, the central step may consist of a C^{3}NOT gate.

The last case is when k is more than or equal to $c - 2$. According to Lemma 4, the C^{c}NOT gate can be decomposed into Toffoli-depth $2\lceil log_2 c \rceil$-1 circuit.

In fact, there is a logical error in this previous method [2]. In other words, even when the number of work qubits increases, Toffoli-depth of the resulting circuit also sometimes increases. A detailed description of this error is introduced in Sect. 4. In that section, a figure is shown as an example of such an error (Fig. 12).

Another technique for optimizing Toffoli-depth of MPMCT gates utilizes dynamic programming to derive optimal solutions for each stage in the circuit [6]. Some characteristics of the method presented in this study are briefly mentioned. In this study,

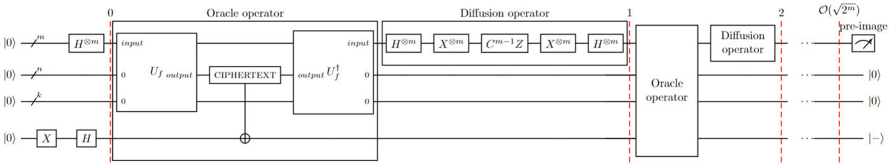

Fig. 8 Grover's algorithm for a given cryptosystem f. A key or pre-image can be found using this quantum circuit with time complexity of $\mathcal{O}(\sqrt{2^m})$. MPMCT gates should be installed in each operator and the type and number of work qubits vary depending on the type of cryptosystem circuit

the case where CWQs and DBQs coexist was not considered, and only cases with one type of work qubits were considered and presented, respectively. In the case of a sub-method considering the case where there are only DBQs, it can be seen as a method that solves the limitation of Lemma 1 because it can handle the case where the number of DBQs is not large enough. In the case of another sub-method where CWQs are considered only, it is designed to ensure that Lemma 4 can be applied by retaining a sufficient number of CWQs in every step. In other words, it is a kind of greedy algorithm designed to use circuits with optimized Toffoli-depth at each step. This is a difference from the former previous technique [2] mentioned above that this former study tried to retain a sufficient number of DBQs in all steps. A more detailed explanation is given in a previous paper [6].

The method presented in this study complements the shortcomings of these two previous studies. The proposed algorithm first solves the logical error of the former previous technique and returns a circuit with a lower Toffoli-depth. In this previous technique, as mentioned above, it is divided into three ranges according to the value of k, but in the proposed technique, it is divided into five ranges. Also, unlike another previous technique [6], it considers both kinds of work qubits in every step. It is explained in detail in the next section.

2.3 Grover's algorithm

Figure 8 illustrates the quantum circuit for Grover's algorithm [12]. To find a m-bit key or pre-image of a given cryptosystem, the algorithm requires $\mathcal{O}(\sqrt{2^m})$ iterations of the Grover iteration, which is composed of the Oracle operator and the Diffusion operator.

The Oracle operator in Grover's algorithm is composed of the quantum circuit U_f for the cryptosystem f, the inverse circuit $U_f^\dagger$, and a comparator. The search space consists of m data qubits, while n output qubits represent output values after passing through the quantum gate U_f that represents the cryptosystem f. (As a side note, if the cryptosystem is an in-place version circuit, these output qubits may not be needed.) k CWQs can help execute the cryptosystem circuits and are initialized through the uncomputation step via the $U_f^\dagger$ gate. When implementing the inverse circuit, the order of the gates in the original cryptosystem's circuit is arranged in reverse order. Finally, one work qubit serves as the Oracle qubit, which is used in the target part of the comparator (a C^{n}NOT gate) in the Oracle operator and is crucial for the phase kick-back technique. To implement the comparator, m+k DBQs can be used, and additional help from CWQs not shown in the figure can also be used. In fact, although the oracle qubit is conventionally employed for the phase kick-back technique in Grover's algorithm, it can be deleted. As shown in [25], a C^{n-1}NOT gate is sufficient for an n-bit ciphertext when implementing the comparator. A C^{n-1}Z gate can be constructed by adding two H gates, and the phase kick-back can be achieved through this resulting C^{n-1}Z gate, as described in Eq. 3. Thus, the desired operation can be accomplished using a C^{n-1}NOT gate instead of a C^{n}NOT gate.

$$C^{n-1}Z : |x_1 \ldots x_n\rangle \rightarrow (-1)^{x_1 \ldots x_n}|x_1 \ldots x_n\rangle \tag{3}$$

In the Diffusion operator, a C^{m-1}Z gate can be used when the size of the key or pre-image M is m bits. The required MPMCT gate varies depending on the size of the pre-image M in the Diffusion operator. This operator is composed of the C^{m-1}Z gate and H and X gates surrounding this MCT gate. To implement the C^{m-1}Z gate, n+k CWQs can be provided.

We used Grover's algorithm for our technique as a concrete example, while the cryptosystems used are SHA-256 or SHA3-256 cryptosystems [3, 11]. As explained now, a C^{n-1}NOT gate is required in the Oracle operator, and a C^{m-1}NOT gate is used in the Diffusion operator. These two MPMCT gates need to be implemented differently since the states of the work qubits are different. Quantum resources of MPMCT gates according to a given number and type of work qubits are summarized in Sect. 4. The implementations of the MPMCT gates are critical to the efficiency of the entire Grover's algorithm because the efficiency of implementing the cryptosystem and MPMCT gates determine the quantum resources required to run the algorithm. A previous study found that the optimal number of Grover iterations is about 0.690 $\cdots \sqrt{2^m}$ [9]. In this work, we will use this formula to calculate the quantum resources required to implement the algorithm.

3 Proposed method

3.1 New lemma

Before presenting the advanced method, we have an observation that may be utilized in our proposed method. This lemma shows the lower bound for T-depth in Lemma 3. The difference from the lemmas presented in the previous section is that it considers CWQ and DBQ simultaneously (Lemma 5).

Lemma 5 *For c≧4, if there is 1 CWQ and c- 5 DBQs available, A C^{c}NOT gate can be expressed with T-depth 4c-5 and Toffoli-depth 4c-10. Obviously, the C^{4}NOT or C^{5}NOT gate does not require any DBQs in this lemma.*

This lemma can be described by decomposing the C^{c}NOT gate into two Toffoli (C^{2}NOT) gates and a C^{c-1}NOT gate. By decomposing the C^{c}NOT gate in this way, c-3 DBQs needed to decompose the C^{c-1}NOT gate can be given for using Lemma 1. Of course, after decomposition with Toffoli gates, T-depth can be reduced to 4c-5

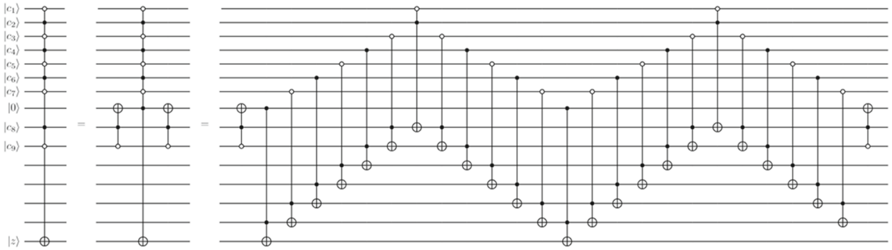

Fig. 9 A C^9NOT gate with 1 CWQ and 4 DBQs. It can be expressed as a circuit with Toffoli-depth 26 and T-depth 31. T-depth of the gate obtained using this lemma may be one less than that of Lemma 1. Moreover, compared to the result of Lemma 3, T-depth is reduced by about 23%

through existing T-depth reduction techniques. Figure 9 provides an example for this lemma by implementing a C^9NOT gate.

It is noted that T-depth of 4c-5 presented in Lemma 4 is smaller than T-depth of 4c-4 in Lemma 1. While Lemma 1 uses three additional DBQs to implement the gate, no CWQs are required. This shows the significant advantage that CWQs can provide in gate design compared to DBQs.

3.2 Our method sketch

We now present our technique. Our work considers the number of CWQs and DBQs to be provided in all parts of the implemented circuit simultaneously, which is an improvement over the previous studies. One of the previous method [2] aimed to maximize the use of CWQs in the front & back steps and DBQs in the central step. In contrast, our proposed method utilizes both CWQs and DBQs in all steps of the circuit. The key idea is to adjust the number of CWQs available to match as much as possible the number of control lines handled in the central step and to choose the number of the sub-MCT gates used in the front & back steps by considering the number of their control lines. With this technique, we can use Lemma 4 instead of Lemma 1 in the central step.

Assume we want to design a C^cNOT gate with k> $\lfloor c/m \rfloor$(+1). Let us suppose that the maximum number of control lines among the sub-MCT gates used in the first stage (of the front step) is m ($\geqq$ 3). We divide the control lines of the C^cNOT gate into $\lfloor c/m \rfloor$+1 groups of control lines, denoted as $C_1, \cdots, C_{\lfloor c/m \rfloor+1}$. Unlike the previous method [2], the number of groups is not determined by the number of CWQs, but by the number of control lines of the sub-MCT gate to be used. We set $|C_1|$ = . . .=$|C_{\lfloor c/m \rfloor}|$=m, and $|C_{\lfloor c/m \rfloor+1}|$ = c-m$\lfloor c/m \rfloor$. At this stage, we use $\lfloor c/m \rfloor$(+1) CWQs as target parts of sub-MCT gates. If $|C_{\lfloor c/m \rfloor+1}|$ is 2 or more, one CWQ is required more (Table 2).

In the central step, we try to decompose the $C^{\lceil c/m \rceil}$NOT gate to Toffoli gates using the remaining $k - \lfloor c/m \rfloor(-1)$ CWQs. If $k - \lfloor c/m \rfloor(-1) \geqq \lfloor c/m \rfloor(+1) - 2$, we can use Lemma 4 in this step. In the front & back step, we can apply Lemma 1 if we have $\lfloor c/m \rfloor(m-2) + max\{c - m\lfloor c/m \rfloor - 2, 0\}$ DBQs. Therefore, we can implement the given MPMCT gate as a circuit with Toffoli-depth approximately equal to $4(m-2) \times 2 + 2\lceil \log_2 \lceil c/m \rceil \rceil - 1$.

Table 2 A C^cNOT gate decomposition in the central step in our proposed method

$\lvert C_{\lfloor c/m \rfloor+1} \rvert$ c-m$\lfloor c/m \rfloor$	Sub-MCT gate $C^{\lceil c/m \rceil}$NOT	Remaining #CWQ k-$\lfloor c/m \rfloor$(-1)
0	$C^{\lfloor c/m \rfloor}$NOT	k-$\lfloor c/m \rfloor$
1	$C^{\lfloor c/m \rfloor+1}$NOT	k-$\lfloor c/m \rfloor$
2 ∼	$C^{\lfloor c/m \rfloor+1}$NOT	k-$\lfloor c/m \rfloor$-1

When C^mNOT gates are used as sub-MCT gates in the front & back steps, the number of control lines in the sub-MCT gates, and the remaining CWQs in the central step are determined as in this table. If the number of remaining CWQs is greater than or equal to the number of control lines in the sub-MCT gate minus 2, then the central part can be designed with Toffoli-depth of approximately $2\lceil \log_2 \lceil c/m \rceil \rceil - 1$, according to Lemma 4

To determine the number of control lines m for the main sub-MCT gates used in the front & back steps, we can refer to Table 2. In this table, the case with the smallest number of remaining CWQs and the largest MCT gates handled in the central step is when $|C_{\lfloor c/m \rfloor+1}| \geq 2$. To ensure optimized T-depth in the central step, we need to satisfy the inequality k-$\lfloor c/m \rfloor$-1 $\gtrapprox$ $\lfloor c/m \rfloor$+1+1. Solving for m, we obtain an expected value for m as m≈2c/(k-3). We can expect that the optimal Toffoli-depth (T-depth) is obtained in the central step when the value of m is approximately 2c/(k-3).

If the number of DBQs is not equal or more than $\lfloor c/m \rfloor (m-2) + max\{c - m\lfloor c/m \rfloor - 2, 0\}$, then $\lfloor c/m \rfloor$ C^mNOT gates cannot be installed in the front step when considering lemma 1. At this time, it is tried to install $\lfloor d/(m-2) \rfloor$ C^mNOT gates instead. Lemma 1 can be applied to these $\lfloor d/(m-2) \rfloor$ sub-MCT gates. The sub-MCT gate installation strategy according to the various number of work qubits is described in detail in "Appendix B."

3.3 Our method's process

In our method, the design for a given C^cNOT gate involves four steps: front, central, back, and additional reduction steps. Steps may consist of multiple stages for achieving lower Toffoli-depth. Only the central step is composed of one stage because it is a step to which one of the lemmas is applied. Different values of m may be selected for each stage. As explained earlier, one of the lemmas is selected and used in the central step. Lemma 1 is used for sub-MCT gates by default in the front & back steps, but further reductions may be possible. Our method goes through the following process. We repeat the process steps for values of m from c-1 to 2.

1. **Front step.** If $d \geq \lfloor c/m \rfloor (m-2) + max\{c - m\lfloor c/m \rfloor - 2, 0\}$ then set l = min{ $\lfloor c/m \rfloor$ + H(c-m$\lfloor c/m \rfloor$-2), k}.[1] Otherwise, set l = min{ $\lfloor d/(m-2) \rfloor$+1,k}. To divide the control lines into l groups, we set $|C_1|$ = ... = $|C_{l-1}|$ = m and $|C_l|$ = c-m(l-1) if c-m$\lfloor c/m \rfloor$ $\geq$ 2. In the front step, a stage may be constructed using $l-1$ C^mNOT gates and one single sub-MCT gate of which the number of controls is less than m by applying Lemma 1. If the work qubits required to implement these sub-MCT

[1] H(x) is a Heaviside function. Namely, H(x) = 1 if x $\geq$ 0 or 0 otherwise.

gates are insufficient, the process stops for this m and proceeds for the following m. *l* CWQs are used as target lines for these sub-MCT gates. Toffoli-depth for this stage may be 4(m-2). The number of available CWQs in the next stage (or step) is reduced by the number of sub-MCT gates in the previous stage right before. *d* increases by the number of control lines of sub-MCT gates. The number of control lines dealing with in the next stage (or step) decreases by the number of controls handled in the previous stage and increases by the number of installed sub-MCT gates. If the remaining number of CWQs is small or large enough to use one of the lemmas, go to the next step. Otherwise, another stage is formed recursively.

2. **Central step.** In the central step, one of the lemmas is used and applied. The control lines of the sub-MCT gate handled in this step are the target lines of the previous stage right before. The target line is that of the original MCT gate. The number of DBQs is increased by the number of controls handled in the front step, and the number of CWQs is decreased by the number of sub-MCT gates installed in the front step. If no CWQs are available and the number of DBQs is not good enough, the technique dedicated to DBQs [6] is considered. If enough CWQs are left to use Lemma 4, Toffoli-depth can be expressed in logarithm form in this step.
3. **Back step.** The back step is composed of the reverse order for the gates used in the front step. If the recursive way is used in the front step, the back step also consists of several stages.
4. **Further reduction step.** If there are enough CWQs or DBQs, then further reductions could be attempted in the front & back steps. If there are more than $2\lfloor c/3 \rfloor$ CWQs in the first place, it is better to apply Lemma 4 instead of Lemma 1. Also, if the DBQs are large enough that some do not need to be used in other stages, Toffoli-depth for these two steps might be lowered from (2m-3)×2 to 2 m-3×2 by canceling some Toffoli gates.
5. **Comparison of T-depth values.** Toffoli-depth and T-depth of the created circuit are recorded for each m. Since the recursive way is used, more than c-2 results may be recorded. If we encounter a situation where both Toffoli-depth and T-depth values increase as value m decreases, the process terminates before m equals 2. We compare T-depth values recorded to select the best T-depth decomposition way from the list. We select the way among approximately c-2 decompositions recorded in the list that makes the lowest T-depth.

The proposed technique examines the cases as many as possible until m becomes 2 or the algorithm is stopped to ensure completeness and to include the results of the previous studies in the list. Toffoli-depth (T-depth) value of the resulting circuit for each m is recorded in the list. It will be observed that these values gradually decrease and then rise again. Toffoli-depth/T-depth recording is performed each time a resulting circuit is made, and as m decreases, the values also may decrease. However, at some point, both values may rise. The decomposition design way for achieving the best T-depth before reaching that point is already included in the list so the process may be stopped before m becomes 2.

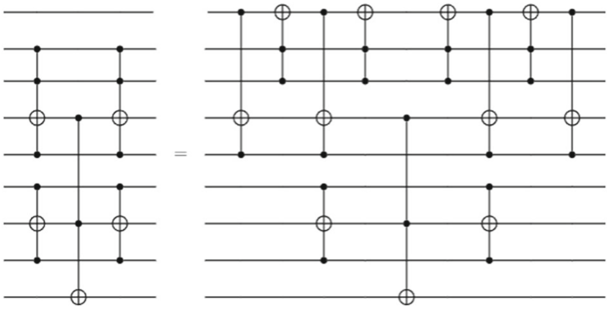

Fig. 10 A C^5NOT gate with 2 CWQs and 1 DBQ. In the front & back steps, Toffoli-depth can be additionally reduced by 2 and T-depth by 4

When designing a technique based on the core ideas mentioned, it is critical to make an acceptable difference between the remaining number of CWQs and the control lines of the sub-MCT gate being handled in the central step. Lemma 4 cannot be directly applied when the number of CWQs remaining is less than the control lines by three or more in the central step. In that case, the front step tends to perform recursively again, so the front step might consist of multiple stages.

When executing the process presented above, it is necessary to check whether it is possible to additionally reduce Toffoli-depth (T-depth) in front & back steps. Let us consider a scenario where l sub-m-controlled-NOT gates are installed in a stage (in the front or back step) and l CWQs are utilized in target parts. In such a case, if there are additional l(m-2) CWQs, then each sub-gate can be implemented as a circuit with Toffoli-depth $2\lceil\log_2 m\rceil$-1. Some readers may wonder when this Lemma 4 can be applied in the front & back steps. As mentioned later, when $2\lfloor c/3\rfloor \lessapprox k \leqq c-3$, Lemma 4 can be applied in the front & back steps. Using Lemma 4 in these steps is consistent with the approach taken in [6] that considers CWQs only.

Meanwhile, if there are enough DBQs available, Toffoli-depth can be further reduced by canceling gates between the front and back steps where Lemma 1 is applied. When there are approximately l(m-2) DBQs, corresponding stages in these steps may be implemented in the same form using Lemma 1. If these DBQs used in these stages do not have to be used in other stages, some Toffoli gates made in the front & back step can cancel each other, resulting in a reduction of Toffoli-depth from $2x4(m-2)to2x(2m-3)$. An example of reducing Toffoli-depth by 1 each in the front & back steps due to the help of 1 DBQ is shown in Fig. 10. Since two Toffoli gates are placed consecutively, they can be deleted.

3.4 Extreme case) #CWQs k is too small

We tried to predict which value m will lead to the optimal Toffoli-depth (T-depth) when the number of given CWQs is too small. This is because equation 2c/(k-3) cannot be used when k is less than 4. For the case where k = 0 or 1, MPMCT gate decomposition can be attempted using the lemmas mentioned earlier. When k = 2, all the available CWQs are attempted to be used in the front step, so the expected value of

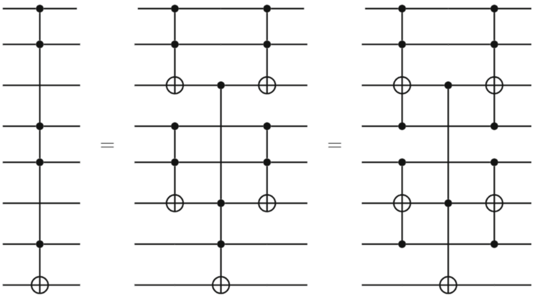

Fig. 11 A C^{5}NOT gate with 2 CWQs. The circuit subfigure in the middle corresponds to the case where $m = 2$, and the circuit subfigure on the right where $m = 3$. An existing target qubit can be used as a DBQ in front and back steps

Table 3 Various C^{c}NOT gate decompositions

#control	#CWQ	#DBQ	T-depth	Toffoli-depth	Remark
4	0	2	12	8	Lemma 1
6	0	1	28	16	Lemma 2
4	1	0	12	6	Lemma 3
9	1	0	40	32	Lemma 3
12	10	0	10	7	Lemma 4
9	1	4	31	26	Lemma 5
5	2	0 or 1	12	7	Proposed method
6	3	2	11	6	Proposed method

This is a list of simple examples mentioned in this paper

m is approximately equal to or less than c/2. In the case of $k = 3$, note the possibility of having optimal Toffoli-depth in the central step, even if optimal T-depth cannot be achieved. So we use the alternate formula k - $\lfloor c/m \rfloor$-1 $\lessapprox$ $\lfloor c/m \rfloor$+1-2 to obtain optimal Toffoli-depth. Therefore, we have the formula m = $\lceil 2c/3 \rceil$. Of course, since these values do not consider the number of DBQs at all, so if the number of DBQs is too small, the optimal T-depth value may not be obtained from the expected values for m.

Consider Fig. 11 as an example where we attempt to decompose the C^{5}NOT gate using only 2 CWQs. Since $\lceil c/2 \rceil$ = 3, we try to decompose it for m = 3. When m = 3, it can be implemented with Toffoli-depth 7 and T-depth 12 circuit. Note that an existing target qubit can also be utilized as a DBQ in front & back steps.

As another example, let us decompose a C^{6}NOT gate with the help of 3 CWQs and 2 DBQs. Since k=3, it can be expected that a C^{4}NOT gate will lead to an optimal circuit based on the equation $\lceil 2c/3 \rceil$=4. However, As a result, we can implement a circuit with Toffoli-depth 6, and T-depth 11 when m = 2. The specific examples discussed in this paper are summarized in Table 3.

3.5 Our strategy depending on #CWQs k

In the previous study [2], the technique was presented by dividing the range of the number of CWQs k into three categories, based on the number of controls of the desired MPMCT gate. However, in this work, the range for k is tried to divide into five categories. When k is near the boundary value of each interval, both cases of nearby categories are recommended to consider. Again, it is important to consider not only the number k of CWQs, but also whether there are enough DBQs available to apply Lemma 1 in the front & back steps.

- $k = 0$. If no CWQs are available, Lemmas 1 and 2 are considered for circuit implementation. For cases where the number of DBQs is less than c-2 and greater than 1, the method using only DBQs from the previous study [6] is considered.
- $1 \leqq k \lessapprox 4$. For the case when k=1, either Lemma 3 or 5 can be utilized. Because the number of CWQs is too small, the formula m $\lessapprox$ 2c/(k-3) cannot be used, so the alternative formulas mentioned above are used to expect the lowest point for T-depth when carrying out the proposed process. If there are enough DBQs, additional reductions in Toffoli-depth may be possible in the front & back steps. If DBQs exist insufficiently, the previous method [6] that solely considers DBQs may have to be used in the central step.
- $4 \lessapprox k \lessapprox 2\lfloor c/3 \rfloor$. This is the range used to apply the process presented above as it is. In general, an optimal circuit can be expected to obtain when m is approximately equal to 2c/(k-3). However, a circuit with optimized T-depth is sometimes obtained when the process is applied recursively. In addition, in some cases, the best results could be achieved even when Lemma 1 is used in the central step. Also, since Lemma 1 is highly likely to be used in front & back steps, we check whether an additional reduction based on DBQs is possible.
- $2\lfloor c/3 \rfloor \lessapprox k \leqq c - 3$. An optimized decomposition can be obtained when the number of control lines m for main sub-MCT gates in the front & back steps is 3. Additionally, since the number of CWQs is sufficiently large, Lemma 4 can be applied in all steps like the method considering CWQs only in [6]. It is observed that the optimized Toffoli-depth (T-depth) remains constant even as the value of k increases in this range.
- $c - 2 \leqq k$. This is the interval where Lemma 4 can be applied, which corresponds to the last interval in the previous study [2].

Some readers may wonder whether the presented categories can be further subdivided. For instance, they may suggest that the range $4 \lessapprox k \lessapprox 2\lfloor c/3 \rfloor$ could be split into smaller intervals, such as $3\lfloor c/4 \rfloor \lessapprox k \lessapprox 2\lfloor c/3 \rfloor$, $4\lfloor c/5 \rfloor \lessapprox k \lessapprox 3\lfloor c/4 \rfloor$, and so on. However, such cases are rare to in practice, as shown in Lemma 6. In other words, the scenario that we use C^{4}NOT, C^{5}NOT gates, etc., at front & back steps as main sub-MCT gates and apply Lemma 4 is frequently unlikely to occur.

Lemma 6 *For c≥0 and c is not equal to 3, 6, 7, or 15, we have* $3\lfloor c/4 \rfloor \geq 2\lfloor c/3 \rfloor$.

This lemma shows a limitation on the possibility of further classification as mentioned earlier. The proof of this lemma is presented in "Appendix C." Meanwhile, in some instances, applying the proposed method recursively or utilizing Lemma 1 in the central step may lead to a circuit with the lowest T-depth, as mentioned above. Conducting further research to identify such cases could help refine the existing classification.

4 Experimental evaluations

4.1 Comparison to the previous method

The first subfigure in Fig. 12 shows the contrast between the previous method [2] and our proposed method. The first subfigure displays T-depth of the circuit that can be implemented based on the number of CWQs provided when implementing the C^{255}NOT gate utilizing 512 DBQs. This graph clearly shows that the previous method exhibits a section in which T-depth increases as the number of CWQs increases, indicating a flaw in the technique [2]. The previous technique had a limitation in that it focused primarily on the number of CWQs and less on the number of DBQs. In particular, the problem arises because the previous method employs an excessive number of CWQs in the front & back part in the circuit, leaving the central part of the circuit with no choice but to use Lemma 1. DBQs are not fully utilized in the front & back part of the circuit and were only heavily used in the central part. Consider the number of CWQs used that can be approximated as $c/2$, $c/2 + c/2^2$, $c/2 + c/2^2 + c/2^3$, and so on, for the number of control lines c in the MPMCT gate. When this previous technique is used, T-depth (Toffoli-depth) of the implemented circuit tends to increase and then decrease before and after the neighborhoods of the numbers of CWQs mentioned. That is, these numbers lead to local extreme values for T-depth. This phenomenon is illustrated well in the first subfigure.

In contrast, the proposed technique achieves a rapid reduction in T-depth as the number of CWQs provided increases. In the proposed technique, if enough CWQs available remain in the central part, then Lemma 4 can be used instead of Lemma 1. This leads to a logarithmic form for Toffoli-depth (T-depth) of the central part of the circuit, instead of linear. The reason for this difference is that the previous method forms the algorithm based on the number of CWQs available in each stage (time slice), while the proposed technique forms the algorithm based on the number of control lines of sub-MCT gates to be used.

The second subfigure shows that there is an overwhelming utility of CWQs over DBQs. T-depth of the created circuit is more correlated with the number of CWQs than with the number of DBQs. It can be experimentally seen that there are no logical errors in our method. Since an MPMCT gate construction is impossible in our setting when (k,d) = (0,0), T-depth value is arbitrarily set to 100,000 in this condition.

In Fig. 13, each subfigure's x-axis represents the number of DBQs. In the first figure, the results of the previous method and our method are shown together. Even though we added our further reduction process to the previous method, there is a difference from the results. When the number of CWQs is small, the number of DBQs may affect T-depth value more. Second graph shows that even if the number of CWQs does not change and is small, further reduction in T-depth may be possible when the number of DBQs is sufficiently large in our further reduction step.

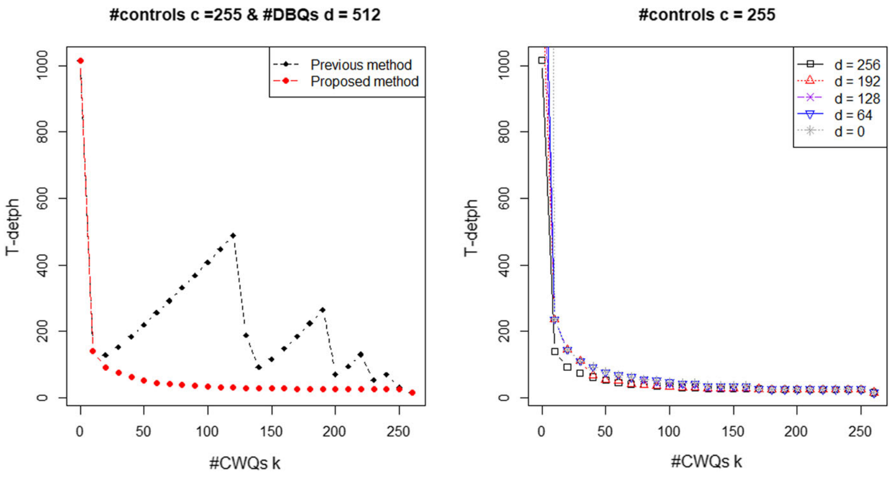

Fig. 12 A C^{255}NOT gate construction with the different number of work qubits. The first subfigure compares the results between the previous method [2] and our method. As the number of CWQs increases, we observe a radical decrease in T-depth, indicating the efficiency of our approach. In contrast, the previous method was found to have some logical errors. The second subfigure experimentally shows that our method is much more affected by CWQs than DBQs. It can also be confirmed that no logical errors occur in our method

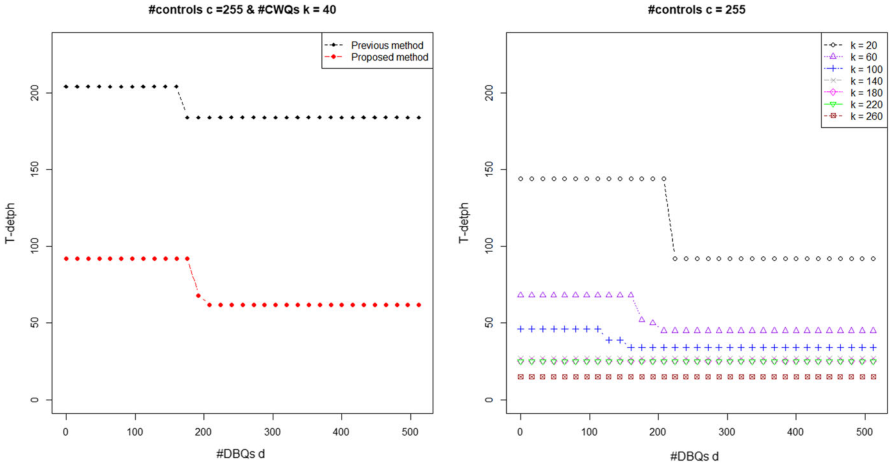

Fig. 13 Other graph versions for a C^{255}NOT gate construction with the various number of work qubits. They show that when the number of CWQs is fixed and small, T-depth may be reduced through a further reduction process as the number of DBQs increases

4.2 Application to Grover's algorithm

As mentioned in the previous Sect. 2, we utilized our proposed technique to implement the MPMCT gates necessary for Grover's algorithm while employing the SHA-256 and SHA3-256 cryptosystems. An (n-1)-controlled-NOT gate is placed for an n-bit ciphertext in the Oracle operator, and a (m-1)-controlled-NOT gate is used in the Diffusion operator when the size of the key (or pre-image) M is m bits. Therefore the required MPMCT gate varies depending on the size of the pre-image M in the Diffusion operator. When constructing Grover's algorithm based on the SHA-256 circuit, we set the message space to 2^{266} and 2^{447}, respectively. 266-bit is the message length selected in a previous study [9], while 447-bit is the maximum size of the original message that can be processed in one message block in SHA-256. The reason they chose 266-bit was that they thought it was an appropriate value for the aforementioned optimal number of Grover iterations.

We also calculated quantum resources for the MPMCT gates required to implement Grover's algorithm for SHA3-256. We cover the message sizes of 266 bits and 1086 bits, one of which is the maximum length of an original message that can be processed within a message block in SHA3-256, so we need to implement either the C^{265}NOT or C^{1085}NOT gate in the Diffusion operator. Since both SHA-256 and SHA3-256 have a bit length of 256 bits in the ciphertext, the Oracle operator requires the C^{255}NOT gate in both cases.

Table 4 shows the location of the MPMCT gate used in Grover's algorithm and compares the quantum resources (Toffoli-depth & T-depth) required for MPMCT gates between the previous method [2] and our proposed method. The number of DBQs (#DBQs) and CWQs (#CWQs) available can vary depending on the version of the hash function circuit and the length of the message. As an example, let us consider the SHA-256-Z1 circuit with Width of 768 qubits. After passing through this cryptosystem circuit in Grover's algorithm, 256 qubits form an ciphertext space, while the remaining 512 qubits exist as superposed values formed by the message schedule algorithm. That is, 512 DBQs can assist in building the C^{255}NOT gate, while CWQs do not exist at the point. Since there are more than 253 DBQs, Lemma 1 can be applied, and thus the C^{255}NOT gate can be implemented as a circuit with Toffoli-depth 1012 (and T-depth 1016). Since Lemma 1 is used in both methods, there is no difference in the results.

At this time, we use SHA-256-Z3 and Z4 circuits for comparison. After passing these circuits in the Oracle operator, the number of CWQs that can be provided to the MPMCT gate design is 159 and 194, respectively. Note that 159 is less than $c/2+c/2^2$ ≈191 while 194 exceeds this value. When the previous technique is applied with 159 CWQs, the C^{255}NOT gate is implemented as Toffoli-depth 138 circuit. On the other hand, when 194 CWQs are used, the circuit is implemented with Toffoli-depth of 164. As mentioned earlier, the result using 159 CWQs is better than using 194 CWQs in the previous technique [2]. Because too many CWQs were used in the front & back steps, insufficient CWQs are provided in the central part of the circuit. Only Lemma 1 can be used in the central part, resulting in a not-efficient circuit. On the other hand, when using our method, the results have no logical error. Also, better results than those of previous studies are presented.

A C^{265}NOT or C^{446}NOT gate is implemented in the diffusion operator. After passing through the Oracle operator, at least 256 CWQs can be provided because the qubits forming the ciphertext have been initialized. For a message length of 447 bits, 65 CWQs can be used more, and if the message M is 266 bits long, 246 CWQs can be provided more. When the message length $|M|$ is 266, the number of available CWQs is sufficient to implement the C^{265}NOT gate with the same using Lemma 4, as the previous method. For $|M| = 447$, the cases for SHA-256-Z1 and Z2 show that using

Table 4 Quantum resource comparisons for MPMCT gate between results of the previous method and those of the proposed method [2, 3]

Secure Hash algorithm	Grover's algorithm	$\|M\|$	#controls	#CWQs	#DBQs	Toffoli-depth (T-depth)	
						Previous method [2]	Proposed method
SHA-256-Z1,Z2,Z3,Z4 [3]	Diffusion operator	266	265	502↑	0	17(17)	17(17)
SHA-256-Z1	Diffusion operator	447	446	321	0	394(408)	21(27)
SHA-256-Z2	Diffusion operator	447	446	350	0	100(114)	21(27)
SHA-256-Z3,Z4	Diffusion operator	447	446	480↑	0	17(17)	17(17)
SHA-256-Z1	Oracle operator	–	255	0	512	1012(1016)	1012(1016)
SHA-256-Z2	Oracle operator	–	255	29	512	164(176)	61(74)
SHA-256-Z3	Oracle operator	–	255	159	512	138(152)	21(27)
SHA-256-Z4	Oracle operator	–	255	194	512	164(178)	19(25)
SHA3-256-v1,v2,v3,v4 [11]	Diffusion operator	266	265	1334↑	0	17(17)	17(17)
SHA3-256-v1	Diffusion operator	1086	1085	514	0	2056(2068)	39(47)
SHA3-256-v2,v3,v4	Diffusion operator	1086	1085	1154↑	0	21(21)	21(21)
SHA3-256-v1	Oracle operator	–	255	0	1344	1012(1016)	1012(1016)
SHA3-256-v2,v3,v4	Oracle operator	–	255	640↑	1344	15(15)	15(15)

Depending on the version of the SHA-256 or SHA3-256 circuits, the length of the message $|M|$, and the number of controls for MPMCT gates required within each operator, the number of work qubits that can be provided is different. It can be observed that the proposed method makes more efficient implementation of MPMCT gates. An arrow pointing up stands for 'or more'. More specifically, in the Diffusion operator for SHA-256-Z3 and Z4, the number of CWQs available is 480 and 515, respectively

Table 5 Quantum resources for Secure Hash Algorithms and Grover's algorithm where the length $|M|$ of the message is 266

Secure Hash algorithm	Width	Toffoli-depth for secure Hash algorithm	Toffoli-depth for Grover's algorithim
SHA-256-Z1 [3]	768	32,895	$1.4070... \times 2^{143}$
SHA-256-Z2 [3]	797	12,023	$1.0160... \times 2^{142}$
SHA-256-Z3 [3]	927	6914	$1.1679... \times 2^{141}$
SHA-256-Z4 [3]	962	4418	$1.4946... \times 2^{140}$
SHA3-256-v1 [11]	1600	168	$1.8396... \times 2^{137}$
SHA3-256-v2 [11]	2240	144	$1.7245... \times 2^{135}$
SHA3-256-v3,v4 [11]	4800	96	$1.2075... \times 2^{135}$

When extending the existing cryptosystem circuit to Grover's algorithm circuit, there is no need to add qubits. It can be checked that the proposed approach requires fewer quantum resources to attack cryptosystems compared to the previous study [9]

the proposed method can make better implementations for MPMCT gates in the Diffusion operator. In the case of SHA-256-Z1, the number of CWQs available is 321, which is about 125 less than the number of control lines. The table shows that the proposed method reduces T-depth by about 94%, compared to the previous scheme.

We also calculated the quantum resources required for a pre-image attack algorithm on four versions of the SHA3-256 cryptosystem. In most cases, the number of CWQs or DBQs is sufficiently large that the values are the same as presented in the previous method. The superiority of the proposed method can be confirmed when using the SHA3-256-v1 circuit, which uses a relatively small number of work qubits.

Additionally, we calculated the total quantum resources required to implement Grover's algorithm for Secure Hash Algorithms (Table 5). For comparison with the previous study [9], we calculated Width and Toffoli-depth required when the message length is 266. The number of Grover iterations followed Proposition 3 in [9], which repeats $0.690...\cdot\sqrt{2^n}$ times when the size of the ciphertext is n bits. Since we do not add the Oracle qubit, Width required for Grover's algorithm is the same as that for the cryptosystem implementation. Compared to Table 7 of the previous study [9], we observe that the quantum resources required for the quantum pre-image attack on SHA-256 are reduced significantly in terms of space-time complexity.

5 Conclusion

We present an advanced method than the previous methods for MPMCT gate decomposition [2, 6]. The key idea is to manage the number of available CWQs so that Lemma 4 can be applied in the central part of the circuit. The circuit with the lowest T-depth is selected by trying all possible cases for the main sub-MCT gates used in the front & back parts. Additional Toffoli-depth reduction between front & back parts of the circuit can also be considered if DBQs remain sufficient when Lemma 1 is applied. If CWQs exist enough, Lemma 4 may be applied throughout the circuit.

Compared to the previous method [2], T-depth value of the resulting circuit decreases more rapidly as the number of CWQs increases. Also, the logical fallacy in this previous method has been eliminated. As a concrete example, quantum resources for MPMCT gates used in Grover's algorithm are presented. Since the number of work qubits (CWQs, and DBQs) that can be provided varies depending on the used cryptosystem circuit, MPMCT gate decomposition is also performed using different sub-MCT gates. Additionally, we calculated the total quantum resources (Width, Toffoli-depth) for Grover's algorithm. Quantum resources required for the pre-image attack are reduced significantly compared to the previous study [9].

As mentioned above, there are some cases where an optimal circuit comes out when a different m value is used, which is not the expected value for the control lines of the main sub-MCT gates. Also, there are cases where the optimal circuit returns when Lemma 1 is used in the central part. That is why a list for every possible m is written in our method. Investigating when these exceptions occur could be a future area of research. Through this investigation, a more advanced MPMCT gate decomposition technique considering FTQC at the logical level may be developed.

Author Contributions Jongheon Lee wrote the manuscript. Yousung Kang, You-Seek Lee, Boheung Chung, and Dooho Choi coordinated the project. Dooho Choi reviewed & edited the manuscript. All authors revised and approved the content of the manuscript.

Funding This work was partly supported by Institute of Information & communications Technology Planning & Evaluation(IITP) grant funded by the Korea government(Ministry of Science and ICT(MSIT)) (⟨Q|Crypton⟩, No.2019-0-00033, Study on Quantum Security Evaluation of Cryptography based on Computational Quantum Complexity), by Quantum Computing based on Quantum Advantage challenge research through the National Research Foundation of Korea (NRF) funded by the Korean government(MSIT) (RS-2023-00256221), and by the MSIT, Korea, under the ITRC (Information Technology Research Center) support program (IITP-2023-RS-2022-00164800) supervised by the IITP.

Availability of data and materials Data related to this paper is available on request from one of the authors, Jongheon Lee.

Declarations

Conflict of interest We have no competing interests to declare that are relevant to the content of this article.

Code availability The software implementation for the suggested algorithm used to produce the circuits with optimized Toffoli-depth is not publicly available.

Appendix A: Optimized Toffoli-depth for an MPMCT gate with sufficiently many CWQs

The MPMCT gate can be implemented with Toffoli-depth $\mathcal{O}$(log c) when the number of CWQs is as large as the number of controls c in the gate (Lemma 4). Specifically, for c-2 CWQs, a C^cNOT gate can be implemented as a circuit with Toffoli-depth of $2\lceil log_2 c\rceil$-1, Toffoli-count of 2c-3, T-depth of $2\lceil log_2 c\rceil$+2, and T-count of 8c-9. Previous studies have not provided proof of this lemma and omitted it, hence it is presented here [2, 5].

The proof process presented below can be easily understood by considering a tournament competition. The control lines of the C^cNOT gate are paired by two, just like the teams in a tournament bracket. Each time Toffoli gates are applied, the number of control lines is halved until the central stage is reached, where only 2 control lines remain (as shown in Fig. 7). At this point, the central step can be implemented with a single Toffoli gate. Because of this implementation way, the C^cNOT gate's Toffoli-depth value is $\mathcal{O}$(log c). If c teams compete in a tournament competition, a total of c-2 matches are required to reach the final. This is because c-2 teams lose and are eliminated from the tournament. Therefore, c-2 CWQs are required to implement each Toffoli gate in the front step. After the central stage, an uncomputation step is performed to initialize these c-2 CWQs. Taking into account this uncomputation step, it is easy to see that the Toffoli-count is 2c-3. T-count of the C^cNOT gate circuit can be reduced by replacing 2c-4 Toffoli gates with C^2(-iZ) or C^2(iZ) gates. As a result, the overall T-count is 8c-9. Toffoli-depth of the circuit can be shown to be $2\lceil log_2 c\rceil$-1 through the following process.

Claim 1 *For C^cNOT gate decomposition with c-2 CWQs, Toffoli-depth (or the number of stages) in the front step is i=$\lceil\log_2 c\rceil$-1.*

Proof of Claim 1)

$$\lceil\frac{c}{2^i}\rceil = 2 \Leftrightarrow 2 \geq \frac{c}{2^i} > 1 \Leftrightarrow \log_2 c > i \geq \log_2 c - 1$$

Case 1) c $= 2^k (k \in \mathbb{N})$

$$\log_2 c > i \geq \log_2 c - 1 \Leftrightarrow k > i \geq k - 1 \Leftrightarrow i = k - 1 = \log_2 c - 1 = \lceil\log_2 c\rceil - 1$$

Case 2) c $= 2^{k+\alpha} (k \in \mathbb{N}, 0 < \alpha < 1)$

$$k + 1 > \log_2 c > i \geq \log_2 c - 1 > k - 1 \Leftrightarrow i = k = \log_2 c - \alpha = \lfloor\log_2 c\rfloor$$
$$= \lceil\log_2 c\rceil - 1$$

□

In the front step, a total of c-2 CWQs are used as the target parts of c-2 Toffoli gates. The number of stages (time slices) comprising this step is i=$\lceil\log_2 c\rceil$-1. That is, Toffoli-depth before reaching the central stage is i=$\lceil\log_2 c\rceil$-1. Therefore, with $c - 2$ CWQs, the C^cNOT gate can be constructed as a circuit with Toffoli-depth $2\lceil\log_2 c\rceil$-1(=2i+1).

Table 6 Sub-MCT gates installed in the front step when the number of DBQs is large enough

l	c%m	Installed sub-MCT gates	Remaining #controls	Incremented #DBQs
$< k$	$\geq$2	$\lfloor c/m \rfloor$ C^{m}NOT +C$^{C\%m}$NOT	l	c
$< k$	1	$\lfloor c/m \rfloor$ C^{m}NOT	$l+1$	c-1
$< k$	0	$\lfloor c/m \rfloor$ C^{m}NOT	l	c
$= k$	-	l C^{m}NOT	$c - l(m-1)$	ml

The number of CWQs available in the next stage or step is $k - l$. Only in the first subcase, one additional small sub-MCT gate is installed

Appendix B: Sub-MCT gate installation based on the number of work qubits

In this section, we look at the sub-MCT gate installation strategy, which divides the number of control lines of a given MCT gate according to the number of DBQs and CWQs, and installs them in one stage of the front step. The numbers of remaining CWQs & control lines that will be addressed in the next stage or step are determined, respectively, through this installation strategy. When a given MPMCT gate is a C^{c}NOT gate, the number m of control lines of main sub-MCT gates installed in each stage is set between c-1 and 2. When c and m are determined, it is divided into two cases according to the number of DBQs d. (The number of CWQs is denoted by k.)

Case 1) d $\geq \lfloor c/m \rfloor$**(m-2)+max**$\{$**c%m-2,0**$\}$. We set $l = \min\{\lfloor c/m \rfloor + H($c%m-2$, k)\}$ where $c\%m$ is the remainder after dividing c by m. The partition for control lines is largely divided into four subcases (Table 6). The number of CWQs used in all subcases is equal to l.

In each case, l CWQs are used, so the number of CWQs usable in the next stage or central step is $k - l$. l(-1) C^{m}NOT gates are installed, and an additional single C$^{c\%m}$NOT gate is added in the first subcase. Since the number d of DBQs is large enough, all installed sub-MCT gates can be implemented through Lemma 1. As explained in the main text, if $k - l \geq l(+1) - 2$, Lemma 4 can be applied in the central step, so Toffoli-depth of the resulting circuit can be approximately 4(m-2)x2+2$\lceil log_2 \lceil c/m \rceil \rceil$-1.

Case 2) d $<$ $\lfloor c/m \rfloor$**(m-2)+max**$\{$**c%m-2,0**$\}$. If there are not enough DBQs, $\lfloor c/m \rfloor$ C^{m}NOT gates cannot be installed. We set l = min$\{\lfloor d/(m-2) \rfloor$-1, $k\}$. The control lines of a given MPMCT gate are split into $|C_1| = \ldots = |C_{l-1}| = m$, and $|C_l| = c - m(l-1)$. $(m-2)(l-1)$ DBQs helps when l-1 C^{m}NOT gates are installed by Lemma 1. It is divided into three subcases according to the value of l and the value of $d - (m-2)(l-1)$, which is the number of remaining DBQs (Table 7).

For all subcases, l CWQs are used. For $d - (m-2)(l-1) \geq 1$, additionally a single C$^{d-(m-2)(l-1)+2}$NOT gate or one C^{m}NOT gate is installed at that stage. On the other hand, if d-(m-2)(l-1)=0, an additional single C^{2}NOT gate is installed at that stage. For these three subcases to make sense, three claims about the size of $|C_l|$ must be established. Only then can one additional sub-MCT gate be installed for each subcase.

Claim 2 *$c - m(l-1) \geq d - (m-2)(l-1) + 2$ where $l = \lfloor d/(m-2) \rfloor + 1$ and $d - (m-2)(l-1) \geq 1$.*

Table 7 Sub-MCT gates installed in the front step when the number of DBQs is not large enough

l	d-(m-2)(l-1)	Installed sub-MCT gates	Remaining #controls	Incremented #DBQs
$= \lfloor d/(m-2) \rfloor$+1	$\geq$ 1	(l-1) C^{m}NOT +C$^{d-(m-2)(l-1)+2}$NOT	$c - d - (l-1)) - 1$	$d + 2(l-1) + 2$
$< \lfloor d/(m-2) \rfloor$+1	$\geq$ m-2	(l-1) C^{m}NOT+C^{m}NOT	$c - (m-1)l$	ml
$= \lfloor d/(m-2) \rfloor$+1	0	(l-1) C^{m}NOT+C^{2}NOT	c-(m-1)(l-1)-1	$m(l-1)) + 2$

l CWQs are used, and a single sub-MCT gate equal to or smaller than the C^{m}NOT gate is installed additionally in all three subcases

Claim 3 $c-m(l-1) \geq m$ *where* $l < \lfloor d/(m-2) \rfloor + 1$ *and* $d-(m-2)(l-1) \geq m-2$.

Claim 4 $c-m(l-1) \geq 2$ *where* $l = \lfloor d/(m-2) \rfloor + 1$ *and* $d-(m-2)(l-1) = 0$.

For these claims above, the proof for Claim 2 is shown. The other two claims can be proved simply in a similar way to Claim 2.

Proof of Claim 2)

$$d < (m-2)\lfloor c/m \rfloor + max\{c\%m-2, 0\} < m\lfloor c/m \rfloor + c\%m = c,$$
$$l = min\{\lfloor d/(m-2) \rfloor + 1, k\}.$$
$$|C_1| = \ldots = |C_{l-1}| = m, \text{ and } |C_l| = c - m(l-1).$$
$$l-1 = \lfloor d/(m-2) \rfloor \leq \lfloor c/m \rfloor.$$

$$\text{Case 1)}\lfloor d/(m-2) \rfloor < \lfloor c/m \rfloor$$
$$\lfloor d/(m-2) \rfloor + 1 \leq \lfloor c/m \rfloor$$
$$c \geq m\lfloor c/m \rfloor \geq m(\lfloor d/(m-2) \rfloor + 1)$$
$$\Rightarrow c - m\lfloor d/(m-2) \rfloor \geq m$$
$$\Rightarrow c - m(l-1) - (d-(m-2)(l-1)+2) > m - m = 0$$

$$\text{Case 2)}\lfloor d/(m-2) \rfloor = \lfloor c/m \rfloor$$
$$1 \leq d-(m-2)(l-1) < max\{c\%m-2, 0\} = c\%m - 2 = c - m(l-1) - 2$$

□

It can be easily observed that when d does not exist enough, if $k \geq c-d-2$, then Lemma 4 can be applied in the central step by the logic of the text. That is, if $d < \lfloor c/m \rfloor$(m-2)+max{c%m -2,0}, $k \geq c-d-2$, $l = \lfloor d/(m-2) \rfloor$+1, and m=3, then Toffoli-depth for the resulting circuit is 8 + 2$\lceil log_2\{c-(m-1)(l-1)-1-H(d\%(m-2)-1)(d\%(m-2))\}\rceil$-1.

Appendix C: Proof of Lemma 6

Lemma 6 mentioned in the main text is as follows.

For c $\geq$ 0 and c is not equal to 3, 6, 7, or 15, we have3$\lfloor c/4 \rfloor \geq 2\lfloor c/3 \rfloor$.

Before proving this lemma, we explain why this lemma came out. Some readers might think there may be cases where the C^{4}NOT gate is used as the main sub-MCT gate in the front & back steps, and lemma 4 is applied. In other words, the case may consider if there are about $3\lfloor c/4 \rfloor$ CWQs, then $\lfloor c/4 \rfloor$ of them are used as target qubits for Toffoli gates, and the remaining $2\lfloor c/4 \rfloor$ of them are used as work qubits for lemma 4. For this case to be used in the third range mentioned above, it should be $3\lfloor c/4 \rfloor < 2\lfloor c/3 \rfloor$. But such cases seldom occur.

Table 8 The comparison between two values, $\lfloor c/3 \rfloor + c\%3$ and $\lfloor c/4 \rfloor + c\%4$, is summarized in table

c	$\lfloor c/3 \rfloor + c\%3$	$\lfloor c/4 \rfloor + c\%4$	Difference
12 m	4 m	3 m	m
12 m+1	4 m+1	3 m+1	m
12 m+2	4 m+2	3 m+2	m
12 m+3	4 m+1	3 m+3	m-2
12 m+4	4 m+2	3 m+1	m+1
12 m+5	4 m+3	3 m+2	m+1
12 m+6	4 m+2	3 m+3	m-1
12 m+7	4 m+3	3 m+4	m-1
12 m+8	4 m+4	3 m+2	m+2
12 m+9	4 m+3	3 m+3	m
12 m+10	4 m+4	3 m+4	m
12 m+11	4 m+5	3 m+5	m

From this table, it is apparent that the inequality mentioned earlier is invalid only for the values of c equal to 3, 6, 7, and 15

Proof

$$\begin{aligned} \text{c}=&3\lfloor c/3 \rfloor + c\%3 =(3-1)\lfloor c/3 \rfloor + \lfloor c/3 \rfloor + c\%3 \\ =&4\lfloor c/4 \rfloor + c\%4 =(4-1)\lfloor c/4 \rfloor + \lfloor c/4 \rfloor + c\%4 \\ &\text{where c\%d is the remainder after dividing c by d.} \end{aligned}$$

From the above expressions, it can be seen that it is enough to show $\lfloor c/3 \rfloor + c\%3 \geq \lfloor c/4 \rfloor + c\%4$. Since c%3=0 or 1 or 2, c = 3k or 3k+1 or 3k+2 for some k. Similarly, c= 4l or 4l+1 or 4l+2 or 4l+3 for some l. We list a total of 12 cases by solving the Diophantine equations for c, and then compare the values of the two expressions in Table 8. The table confirms that the inequality $3\lfloor c/4 \rfloor \geq 2\lfloor c/3 \rfloor$ holds true for the majority of values of c. □

References

1. Abdessaied, N., Amy, M., Soeken, M., Drechsler, R.: Technology mapping of reversible circuits to clifford+ t quantum circuits. In: 2016 IEEE 46th International Symposium on Multiple-valued Logic (ISMVL), pp. 150–155. IEEE (2016)
2. Niemann, P., Gupta, A., Drechsler, R.: T-depth optimization for fault-tolerant quantum circuits. In: 2019 IEEE 49th International Symposium on Multiple-Valued Logic (ISMVL), pp. 108–113. IEEE (2019)
3. Lee, J., Lee, S., Lee, Y.-S., Choi, D.: T-depth reduction method for efficient sha-256 quantum circuit construction. IET Inf. Secur. (2022)
4. Amy, M., Maslov, D., Mosca, M.: Polynomial-time t-depth optimization of clifford+ t circuits via matroid partitioning. IEEE Trans. Comput. Aided Des. Integr. Circuits Syst. **33**(10), 1476–1489 (2014)
5. Selinger, P.: Quantum circuits of t-depth one. Phys. Rev. A **87**(4), 042302 (2013)
6. Baker, J.M., Duckering, C., Hoover, A., Chong, F.T.: Decomposing quantum generalized toffoli with an arbitrary number of ancilla. arXiv:1904.01671 (2019)
7. Lidar, D.A., Brun, T.A.: Quantum Error Correction. Cambridge University Press, Cambridge (2013)

8. Fowler, A.G., Stephens, A.M., Groszkowski, P.: High-threshold universal quantum computation on the surface code. Phys. Rev. A **80**(5), 052312 (2009)
9. Kim, P., Han, D., Jeong, K.C.: Time-space complexity of quantum search algorithms in symmetric cryptanalysis: applying to aes and sha-2. Quantum Inf. Process. **17**(12), 1–39 (2018)
10. Biswal, L., Bhattacharjee, D., Chattopadhyay, A., Rahaman, H.: Techniques for fault-tolerant decomposition of a multicontrolled toffoli gate. Phys. Rev. A **100**(6), 062326 (2019)
11. Lee, J.: A study on t-depth and toffoli-depth reduction techniques for efficient quantum circuit designs and their applications to hash functions. Ph.D. thesis, University of Science & Technology (2023)
12. Grover, L.K.: A fast quantum mechanical algorithm for database search. In: Proceedings of the Twenty-eighth Annual ACM Symposium on Theory of Computing, pp. 212–219 (1996)
13. Shenvi, N., Kempe, J., Whaley, K.B.: Quantum random-walk search algorithm. Phys. Rev. A **67**(5), 052307 (2003)
14. Potoček, V., Gábris, A., Kiss, T., Jex, I.: Optimized quantum random-walk search algorithms on the hypercube. Phys. Rev. A **79**(1), 012325 (2009)
15. Ambainis, A.: Quantum walk algorithm for element distinctness. SIAM J. Comput. **37**(1), 210–239 (2007)
16. Amy, M., Maslov, D., Mosca, M., Roetteler, M.: A meet-in-the-middle algorithm for fast synthesis of depth-optimal quantum circuits. IEEE Trans. Comput. Aided Des. Integr. Circuits Syst. **32**(6), 818–830 (2013)
17. Preskill, J.: Quantum computing in the nisq era and beyond. Quantum **2**, 79 (2018)
18. Cruz, P.M., Murta, B.: Shallow unitary decompositions of quantum fredkin and toffoli gates for connectivity-aware equivalent circuit averaging. arXiv:2305.18128 (2023)
19. Duckering, C., Baker, J.M., Litteken, A., Chong, F.T.: Orchestrated trios: compiling for efficient communication in quantum programs with 3-qubit gates. In: Proceedings of the 26th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, pp. 375–385 (2021)
20. Barenco, A., Bennett, C.H., Cleve, R., DiVincenzo, D.P., Margolus, N., Shor, P., Sleator, T., Smolin, J.A., Weinfurter, H.: Elementary gates for quantum computation. Phys. Rev. A **52**(5), 3457 (1995)
21. Miller, D.M., Wille, R., Sasanian, Z.: Elementary quantum gate realizations for multiple-control toffoli gates. In: 2011 41st IEEE International Symposium on Multiple-Valued Logic, pp. 288–293. IEEE (2011)
22. Nielsen, M.A., Chuang, I.: Quantum computation and quantum information. Am. Assoc. Phys. Teachers (2002)
23. Shende, V.V., Prasad, A.K., Markov, I.L., Hayes, J.P.: Synthesis of reversible logic circuits. IEEE Trans. Comput. Aided Des. Integr. Circuits Syst. **22**(6), 710–722 (2003)
24. Gidney, C.: StackExchange: Creating bigger controlled nots from single qubit, Toffoli, and CNOT gates, without workspace. 2015. https://cs.stackexchange.com/questions/40933/creating-bigger-controlled-nots-from-single-qubit-toffoli-and-cnot-gates-with (2015)
25. Gidney, C.: Why is an oracle qubit necessary in Grover's algorithm? https://quantumcomputing.stackexchange.com/questions/2145/why-is-an-oracle-qubit-necessary-in-grovers-algorithm (2018)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.