Quantum Information Processing (2024) 23:153
https://doi.org/10.1007/s11128-024-04365-2

Toffoli-depth reduction method preserving in-place quantum circuits and its application to SHA3-256

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

Received: 25 January 2024 / Accepted: 27 March 2024

Abstract

When implementing a quantum circuit for a desired reversible function, an attempt is made to design an accurate quantum circuit. Then, the quantum circuit is optimized based on a specific cost function, such as design cost, width (the number of qubits), depth, etc. In particular, if an in-place subcircuit itself can be optimized while maintaining its in-place property, it will be a very useful way to increase efficiency without changing the initial architecture of the entire quantum circuit. Furthermore, since its (clean) work qubits can easily be utilized in subsequent subroutines of the quantum circuit, it has an additional important advantage in terms of width. In this paper, for the first time to the best of our knowledge, we present a global Toffoli-depth reduction methodology for an in-place version reversible circuit in the case that the given input circuit is optimized with Toffoli-count. We mainly introduce a process to optimize the χ internal function block in SHA3-256 to explain our Toffoli-depth reduction approach preserving the in-place property, and hence, well-balanced five χ quantum circuits are induced in terms of its width and T-depth. And then, we apply these five χ circuits to design the entire SHA3-256 cryptosystem. One of the proposed SHA3-256

✉ 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: 18 April 2024

quantum circuits has a width of 1600 and a T-depth of 264, and this shows a result of 50% and 38.9% reduction compared to the previous circuit in terms of width and T-depth, respectively. Other versions of our SHA3-256 circuits just required 10–33 qubits per one T-depth compared to the previous results which require over 1800 qubits per T-depth, and so this means that our SHA3-256 circuits are well-balanced. Finally, we constructed Grover's algorithm circuit using each version that realized SHA3-256. When output circuits for the presented method were used, quantum volume values of Grover's algorithm circuits became 33 and 50% of the value when the input circuit was used.

Keywords Quantum circuit · Toffoli-depth · In-place version circuit · SHA3-256

1 Introduction

Many research projects have been conducted on the synthesis or optimization of quantum reversible circuits [1–12]. An exact circuit that expresses a given reversible function is tried to create at first, and then an optimization process for a specific cost function (or cost metric) is performed. Various resulting circuits may be generated depending on which gate set is used and which cost function is considered to optimize. The gate set used in circuit synthesis is mainly the NCV (NOT, CNOT, Controlled-V, and Controlled-$V^\dagger$) gate set or the NCT (NOT, CNOT, and Toffoli) gate set. The cost function, which is also called QC (Quantum cost), can represent one of the following metrics [1, 8, 13]:

- T-depth which is the number of T gates processed non-parallelly in a Clifford+T-based circuit;
- T-count which is the number of T gates;
- Toffoli-depth which is the number of Toffoli gates processed non-parallelly in a NCT-based circuit;
- Toffoli-count which is the number of Toffoli gates;
- Width which is the number of qubits;
- GC which means the gate count or the number of elementary gates;
- Design cost (DC) which means the design cost of a circuit, that is the sum of the design costs of gates (Each gate's design cost can be determined by the number of NCV gates used for implementation [1].); etc.

T-depth and Toffoli-depth can be well defined concretely as: Given a Clifford+T-based circuit, T-depth is the maximum number of T and $T^\dagger$ gates among critical paths [8]. Given an NCT-based circuit, Toffoli-depth is the maximum number of Toffoli gates among critical paths. (A critical path is a path of maximum length flowing from the input side to the output side for a given circuit [13].)

T gates are representative non-Clifford gates and are known to cost much more and take significantly longer to run than Clifford gates when considering fault-tolerant quantum computation (FTQC) [14]. Toffoli gates are one of the composite gates, and T and $T^\dagger$ gates are usually used to construct these gates. So, cost metrics related to Toffoli gates are often used instead of those for T gates [15].

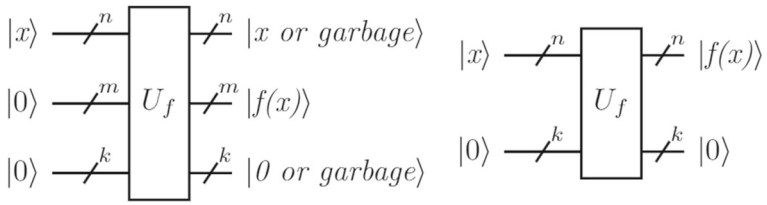

Fig. 1 Out-of-place version and in-place version circuits. For a given (reversible) function f:$\{0,1\}^n \rightarrow \{0,1\}^m$, an out-of-place version circuit tends to return the output values while preserving the input values. Therefore, it is inevitable to use work qubits. In an in-place version circuit, the states of the qubits representing the input values are converted to the output values. The number of (additional) work qubits k could be zero in both versions, but m should not be zero in the out-of-place version. An ancilla-free circuit is an in-place version circuit but an out-of-place version circuit cannot be an ancilla-free circuit. If the circuit is based on the NCT gate set and the reversible function f is an even permutation, the in-place version circuit can be made without work qubits. But if it is an odd permutation, at least one work qubit is needed, so in this case, $k \neq 0$ [17]

Several optimization or reduction techniques have been proposed that consider Toffoli-depth (or Toffoli-count) as a cost function [2–7]. These are based on various techniques such as cut-based balancing, ESOP (Exclusive Sum-of-products) balancing, ESSP (Exclusive sum-of-pseudoproducts) optimization, the pebbling strategy, and DAG (Directed-acyclic graphs)-based approaches such as XAG (Xor-And-Inverter Graphs). They tried to lower Toffoli-depth by making out-of-place version quantum circuits, in which output values are in work qubits and input values tended to remain at the end of the resulting circuit [16]. Out-of-place version circuits are advantageous design forms when input values are still used in subsequent operations (Fig. 1).

These methods are specialized for making out-of-place version circuits, and so an excessive number of garbage qubits (or dirty borrowed qubits) might be produced (Sect. 2). These are not techniques that consider the uncomputation (restoration or clearing) step, which is the process of initializing the state of work qubits, but techniques that allow desired function output values to be written on work qubits, up to swappings. In these previous studies, there has been no approach to output in-place quantum circuits in which no output values appear in work qubits, and all work qubits are initialized. (As a side note, inverse circuits may be utilized to design for conversion from out-of-place to in-place circuits [18].)

In an in-place quantum circuit, input values of data qubits that are not work qubits are converted into output values (Fig. 1). In the figure, the number of (additional) work qubits k could be zero in both versions, but m should not be zero in the out-of-place version. An ancilla-free circuit is an in-place version circuit but an out-of-place version circuit cannot be an ancilla-free circuit. As far as we know, no study has suggested a global Toffoli-depth reduction technique while preserving the in-place property for a given circuit, in which this in-place input circuit is optimized with Toffoli-count.

In this article, we present a global Toffoli-depth reduction methodology while maintaining its in-place property for a given quantum circuit.[1] This input circuit's Toffoli-count is assumed to be optimized. This method is based on the NCT gate set,

[1] This paper is based on Lee's PhD thesis [19]. The PhD thesis contains the contents of a previous work [10] related to T-depth reduction and SHA-256 and this work.

and the key idea is to implement the creation and destruction processes of intermediate values generated in various initial circuits parallel in one single circuit. To achieve this objective, permutation representations for the reversible circuits can be utilized. A more detailed explanation of these can be found in Sects. 2 and 3.

This novel approach has the following two advantages:

- Unlike the previous methods, the proposed method does not use excessively many work qubits but returns a well-balanced circuit for both width and Toffoli-depth (consequently T-depth).
- Since the in-place version circuit is returned, all clean work qubits used are initialized to their original states. Therefore, CWQs (clean work qubits), rather than DBQs (dirty borrowed qubits), can be provided for the next operation design so that the overall circuit can be constructed more efficiently. (Explanations for CWQs and DBQs are shown in Sect. 2.)

This reduction technique is divided into two cases to deal with. One is that Toffoli-count is allowed to increase, and the other is not. Through this reduction technique, we can experimentally confirm that there is a trade-off between Toffoli-count and Toffoli-depth. As a side note, DC and execution time of the resulting circuit may be reduced more by using T-depth reduction techniques [8–10].

To explain our methodology concretely, we first apply on χ internal function, which is used in SHA3-256. As a result, the five χ internal function quantum circuits are provided: The first version is made through an existing reversible circuit synthesis and a (global) Toffoli-count reduction method. This version serves as an input circuit of our presented method. A detailed explanation of this input circuit design process is provided in Appendix A. The second and third versions correspond to the two cases of our presented method, respectively. The last two versions are based on MBQC (measurement-based quantum computation), so intermediate measurement meters exist in the circuits. Quantum AND and $\text{AND}^\dagger$ gates instead of some Toffoli gates are used to compose these circuits [20]. Their Toffoli-depth values are identical to the third and fourth versions' Toffoli-depth values if AND-depth is included in Toffoli-depth. And then, we present five well-balanced SHA3-256 quantum circuits and compare them with the previous results: Each version uses different χ internal function circuits. One of the designed SHA3-256 quantum circuits has width 1600 and T-depth 264, and this shows a result of 50% (38.9%, respectively) reduction compared to the previous work in terms of width (T-depth, respectively). Other versions of our SHA3-256 circuits just required 10–33 qubits per T-depth compared to the previous out-of-place version results which have over 1800 qubits per T-depth, and so this shows that our SHA3-256 circuits are well-balanced. Finally, we designed Grover's algorithm based on SHA3-256 circuits we created. As a result, quantum volume values of the algorithm circuit made with the second and third output circuits were reduced by 66%, and 50% compared to that of the algorithm circuit made with the first input circuit, respectively.

The main contributions of this paper are summarized as follows:

- For the first time, An approach to reduce Toffoli-depth while maintaining the in-place property is tried on a Toffoli-count-optimized quantum circuit.

- We present five well-balanced χ quantum circuits with the in-place property for this approach.
- By applying one of these five χ quantum circuits on SHA3-256 quantum circuit, we reduced its width and T-depth to 1600 and 264. Compared to a previous work [18], this shows 50% and 38.9% reduction in terms of width and T-depth. And other four well-balanced SHA3-256 quantum circuits, with 10, 25, 11, and 33 qubits per T-depth, respectively, are provided. The previous results [3, 6] have 1933 and 1866 qubits per T-depth, respectively.
- We compared the quantum resources required when designing Grover's algorithm circuit for the created SHA3-256 circuits. From quantum volume perspective, much more efficient designs were possible when the output circuits of the proposed method were used than when the input circuit was used.

The rest of the paper is written as follows. In Sect. 2, Toffoli gate is explained, which is the main material used in the proposed technique. In addition, qubits' names depending on their roles are briefly mentioned. We show that specific permutations can correspond to some NCT gates in a quantum circuit. This correspondence relationship will be utilized when choosing several gates and placements when creating initial circuits. In Sect. 3, before the proposed method is introduced, specific examples using χ internal function will be given first to make it easier for readers to understand. For a given input circuit, the output circuit differs depending on whether Toffoli-count is allowed to increment or not. (Other specific examples of the proposed method are listed in 'Appendix B.') In Sect. 4, quantum resource comparisons are presented between SHA3-256 circuits made by the proposed technique and those made in previous studies. Also, quantum resources required when designing Grover's algorithm circuits for the created SHA3-256 circuits are also presented. Finally, we summarize this paper, and future research tasks are presented.

2 Background

2.1 NCT gate set and name of qubit

As mentioned earlier, when trying to synthesize a quantum circuit, it is generally designed based on a set of NCV or NCT gates (Fig. 2). CNOT or Toffoli gates conditionally invert the state of a target qubit. In the case of Toffoli gate, if the states of qubits on the two control lines are 1 ($|1\rangle$), the value of the target qubit is inverted. NOT gate causes a bit-flip on the qubit where the gate is located without any conditions. Toffoli gate can be implemented as a circuit with T-count 7 and T-depth 3 [13]. Our Toffoli-depth reduction technique is applied in the NCT gate set-based quantum circuit.

Toffoli gate can be made active under different conditions using other gates (Fig. 3). It can be made into a gate that operates when the state of one control input is 0 ($|0\rangle$) and the other is 1 ($|1\rangle$). We call this an MPT (a mixed polarity Toffoli) gate. More generally, when the number of control lines exceeds two, it is called a mixed polarity multiple-controlled Toffoli (an MPMCT) gate [21]. These MPT (or Toffoli) gates will

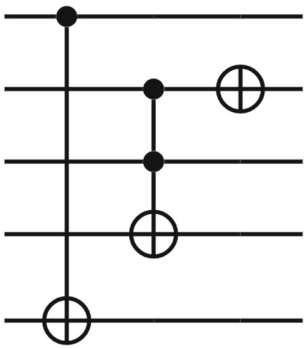

Fig. 2 CNOT, Toffoli, and NOT gates. The more the number of control lines, the less likely the gate is activated. For example, Toffoli gate operates only when the states of the second and third qubits are 1 ($|1\rangle$). Then, a bit-flip occurs at the value of the fourth qubit

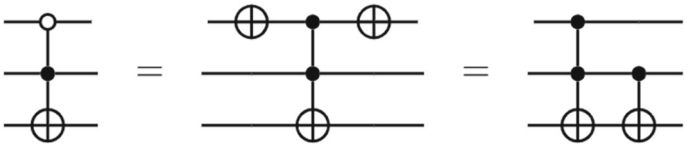

Fig. 3 A mixed polarity Toffoli (an MPT) gate. It can be implemented through one Toffoli gate and two NOT gates or one Toffoli gate and one CNOT gate

be used for the main materials when presenting our Toffoli-depth reduction method. In this paper, Toffoli gates and MPT gates are used as synonyms.

Meanwhile, qubits are called variously depending on how these qubits are used when operations are performed in quantum circuits. In this paper, qubits are classified into three types [15]:

Data qubits exist in quantum circuits as much as the number of Boolean variables used to express a given reversible Boolean function or truth table. That is, these qubits contain values of initial Boolean variables at the front of the quantum circuit and express data information. Our work focuses on designing in-place version quantum circuits so these data qubits have output values at the end of the quantum circuit.

CWQs (clean work qubits) are a type of work qubits (ancilla qubits) that help to perform certain operations in quantum circuits. These qubits' states are known in advance before a particular operation is performed. After a specific operation is performed, they are initialized through an uncomputation step. Normally, $|0\rangle$ are the initial states and they are in separate states at the start of the certain operation. All work qubits used in the circuit built through our proposed method are initialized to CWQs.

Garbage qubits are a type of work qubits, meaning qubits that have not been initialized because there is no uncomputation step while a specific operation is done. Since the values in the qubits are no longer utilized in the remaining operation, they are called garbage qubits. Work qubits that help but are not initialized before this next operation is performed are called DBQs (dirty borrowed qubits). So Garbage qubits can be used as DBQs, not CWQs in the next operation. Operations in a circuit are designed in a more complex way because DBQs are tried to utilize regardless of the values inherent in them. Our proposed technique does not produce any garbage qubits at the end of the circuit.

2.2 Correspondence between permutations and quantum reversible gates

Various reversible circuit synthesis methods using permutation group theory have been proposed [17, 22–28]. In this work, permutation group theory is used as an aid to reduce Toffoli-depth for the in-place version circuit. The idea used is that an operation for an NCT gate can be expressed as a product of specific transpositions (Theorem 1) [29].

Theorem 1 *For n qubits, a reversible operation for an MP-C$^{(k-1)}$NOT (a mixed polarity (k-1)-controlled-NOT) gate can be expressed by a product of 2^{n-k} disjoint transpositions with the Hamming distance 1.*

Basic concepts and theorems about permutation group theory such as transposition can be found in the literature [30]. When there are n qubits (or n wires), the cardinality of all binary representation numbers can be 2^n. An MP-C$^{(k-1)}$NOT gate is a k-qubit gate so there are n-k qubits in the circuit that are not related to this gate. Therefore, this gate can be expressed using 2^{n-k} transpositions. For example, let us look at Fig. 2 presented earlier. CNOT gate is a 2-qubit gate placed on the first and fifth qubits. If the state of the input qubits is $|00001\rangle = |1\rangle$, then the converted state of the (output) qubits is $|10001\rangle = |2^4 + 1\rangle$. Since it is a reversible transformation, it holds in the opposite direction as well. For this gate to work, the value of the first qubit should be 1, and then an inversion occurs in the last qubit. Since there are 3 unrelated qubits for this gate, this operation can be expressed as a product of 2^3 disjoint transpositions. For each gate in Fig. 2, the corresponding products of disjoint transpositions are expressed as follows (1). It can be seen that the Hamming distance between two values in each transposition is 1.

$$
\begin{aligned}
&1)\ \text{CNOT gate: } (00001, 10001)(00011, 10011)(00101, 10101)(00111, 10111)\\
&\quad (01001, 11001)(01011, 11011)(01101, 11101)(01111, 11111)\\
&= (1, 17)(3, 19)(5, 21)(7, 23)(9, 25)(11, 27)(13, 29)(15, 31)\\
&= \prod_{i_1=0}^{2^3-1} (1 + 2i_1, (1 + 2^4) + 2i_1)\\
&2)\ \text{Toffoli gate: } (6, 14)(7, 15)(22, 30)(23, 31)\\
&= \prod_{i_4=0}^{1} \prod_{i_0=0}^{1} ((2 + 2^2) + i_0 + 2^4 i_4, (2 + 2^2 + 2^3) + i_0 + 2^4 i_4)\\
&3)\ \text{NOT gate: } (0, 2)(1, 3)(4, 6)(5, 7)(8, 10)(9, 11)(12, 14)(13, 15)(16, 18)\\
&\quad (17, 19)(20, 22)(21, 23)(24, 26)(25, 27)(28, 30)(29, 31)\\
&= \prod_{i_2=0}^{2^3-1} \prod_{i_0=0}^{1} (0 + i_0 + 2^2 i_2, 2 + i + 2^2 i_2)
\end{aligned}
\tag{1}
$$

An MP-C$^{(k-1)}$NOT gate on n qubits can be identified by the following information.

- $(p_1, \ldots, p_k)$: a set of the positions of the control qubits and a target qubit of the MP-C$^{(k-1)}$NOT gate where $1 \leq p_1 < \cdots < p_k \leq n$
- p_t: a position of the target qubit where $1 \leq t \leq n$.
- $(b_1, \ldots, b_k)$ where for $i \neq t$, $b_i = 0$ if p_i is an off-control position, otherwise $b_i = 1$, and $b_t = 0$.

Therefore, the following notation is sufficient to identify an MP-C$^{(k-1)}$NOT gate on n qubits.

$$\text{MPMCT}[(p_1, \ldots, p_k; p_t), (b_1, \ldots, b_k)], \tag{2}$$

where $1 \leq p_1 < \cdots < p_k \leq n$, $b_i = 0$ or 1, and $b_t = 0$. For example, Toffoli gate in Fig. 2 can be expressed as MPMCT[(2, 3, 4; 4), (1, 1, 0)].

Now, we introduce a new notation for the product of transpositions corresponding to the MPMCT$[(p_1, \ldots, p_k; p_t), (b_1, \ldots, b_k)]$. Firstly, for $2^0 \leq 2^{p_1-1} < \cdots < 2^{p_k}$, consider the following transposition defined by

$$T_I(a, b) := \left(a + i_0 + \sum_{j=1}^{k} i_j 2^{p_j}, b + i_0 + \sum_{j=1}^{k} i_j 2^{p_j} \right), \tag{3}$$

where $I = (i_0, i_1, \ldots, i_k)$ and i_j is a non-negative integer for $j = 0, \ldots, k$. When $I = (0, \cdots, 0)$, $T_I(a, b) = (a, b)$. It is easily checked that the starting transposition of the permutation corresponding to an MPMCT$[(p_1, \ldots, p_k; p_t), (b_1, \ldots, b_k)]$ is

$$(L, L + 2^{p_t-1}), \text{ where } L = \sum_{i=1}^{k} b_i 2^{p_i-1} \tag{4}$$

as you can see the above formula (1) for Fig. 2. Finally, the product of transpositions, $P[(p_1, \ldots, p_k; p_t), (b_1, \ldots, b_k)]$, is defined as follows, and it is evident that its corresponding gate is exactly the MPMCT$[(p_1, \ldots, p_k; p_t), (b_1, \ldots, b_k)]$.

$$\begin{aligned} &P[(p_1, \ldots, p_k; p_t), (b_1, \ldots, b_k)] = \prod_{I \in \Gamma} T_I(L, L + 2^{p_t-1}), \\ &\text{where } L = \sum_{i=1}^{k} b_i 2^{p_i-1}, \text{ and } \Gamma = \{(i_0, \ldots, i_k) : 0 \leq i_0 \leq 2^{p_1-1} - 1, \\ &0 \leq i_j \leq 2^{p_{j+1}-p_j-1} - 1 \text{ for } 1 < j < k, \text{ and } 0 \leq i_k \leq 2^{n-p_k} - 1\} \end{aligned} \tag{5}$$

In formula (5), $T_{(0,\ldots,0)}(L, L + 2^{p_t-1}) = (L, L + 2^{p_t-1})$ is called a *leading-term*. The following corollary is a summarized statement of the above explanation of newly added notations in this paper, and a rephrased statement of Theorem 1.

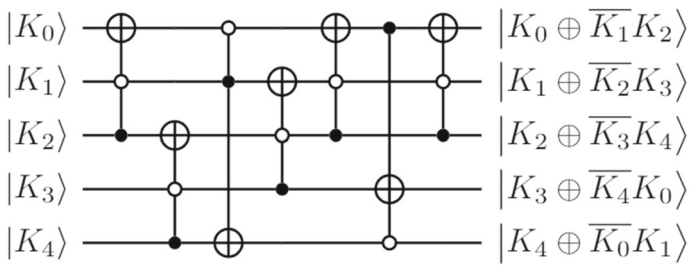

Fig. 4 χ quantum circuit with Toffoli-count 7 (χ-Z1). Toffoli-depth is also 7 and no work qubits are used. For X $\in \{0,1\}$, $\overline{X} = X \oplus 1 = 1 - X$

Corollary 1 *For n qubits, let* $MPMCT[(p_1, \ldots, p_k; p_t), (b_1, \ldots, b_k)]$ *be an MP-*$C^{(k-1)}NOT$ *gate, where* $1 \leq p_1 < \cdots < p_k \leq n$, $b_i = 0$ *or* 1, *and* $b_t = 0$. *Then* $MPMCT[(p_1, \ldots, p_k; p_t), (b_1, \ldots, b_k)]$ *can be expressed by the permutation* $P[(p_1, \ldots, p_k; p_t), (b_1, \ldots, b_k)]$ *defined by* (5), *and its leading-term is* $(L, L + 2^{p_t-1})$, *where* $L = \sum_{i=1}^{k} b_i 2^{p_i-1}$.

As a side note, a SWAP gate can be expressed as a product of 2^{n-2} disjoint transpositions with the Hamming distance 2 in a circuit with width n. This operation, which changes positions between two qubits, may also be used together with NCT gates when synthesizing reversible circuits. This operation can have an impact on reducing DC at the logical level by reducing the number of CNOT gates [22]. Swappings were briefly explained here because SWAP gates are used in the last version among χ circuits to be presented later.

3 For in-place circuits with Toffoli-count optimized, is it possible to reduce Toffoli-depth while maintaining the in-place property?

This section introduces a new approach to reducing Toffoli-depth while maintaining the in-place property. We assume that the input circuit's Toffoli-count is optimized so that the circuit's Toffoli-depth cannot be reduced without the help of (additional) work qubits or local Toffoli-count reduction rules [22, 31–34]. This Toffoli-depth reduction methodology is classified into two cases depending on whether or not the change of Toffoli-count value is allowed. When Toffoli-count increment is allowed, Toffoli-depth can be expected to be more reduced than when Toffol-count is not allowed to increase. In both cases, DC (design cost) increases.

To help readers' understanding, we first explain our approach with χ internal function in detail. The input circuit for our proposed method is seen in Fig. 4. This function is a 5-variable reversible transformation, and since the function is an even permutation, it can be expressed by an ancilla-free circuit. The concrete design process for this circuit can be found in 'Appendix A.' Of course, the synthesis and global Toffoli-count reduction algorithms for making an input circuit may change according to the user's choice. Using this χ input circuit, we show that two version circuits are generated using our method. After that, the proposed method is described in detail.

3.1 Toffoli-depth reduction for χ function in SHA3-256

3.1.1 Case 1: Toffoli-count is not allowed to increase

First, we consider the case of not increasing Toffoli-count. In this case, the key issues are whether to find adjacent Toffoli gate pairs capable of parallel processing and whether the used work qubits can be initialized. It is also important to track changes in the state values of the qubits in the input circuit. As operations are performed in quantum circuits, the state value of each qubit changes, and these state values can be expressed through Boolean expressions. For an in-place version circuit with Toffoli-count optimized, it is too difficult to convert to a circuit with a further reduced Toffoli-depth without adding CWQs. Therefore, Toffoli-depth reduction should be attempted by adding CWQs to this circuit, which means that an uncomputation step that initializes the state of each CWQs should also be implemented in the output circuit.

(1) Try to find candidate pairs We start with the χ-Z1 input circuit in Fig. 4. Because width of χ-Z1 is 5, parallel processing of Toffoli gates is obviously impossible in χ-Z1 circuit without adding work qubits. Looking at the locations of Toffoli gates, it can be confirmed that the fifth MPT gate in χ-Z1 circuit is commutative with the third and fourth MPT gates. Since the on-control line of the fifth MPT gate and the off-control line of the fourth MPT gate are the same, they can be swapped. Also, if this position exchange occurs, then the third MPT gate and the fifth MPT gate are adjacent. Since the on-control part of the third MPT gate and the off-control part of the existing fifth MPT gate share the same line, these two gates are also commutative. Therefore, there are two parallel arrangement candidate pairs, the third and fifth gates, and the fourth and fifth gates in χ-Z1 circuit.

The fact that these pairs are interchangeable can also be confirmed mathematically using permutation. Through the expression (6), it is confirmed that the selected gates have disjoint permutations with each other. We denote that the permutations corresponding to the first, second, third, fourth, and sixth gates in the circuit are P_0, P_2, P_4, P_1, and P_3, respectively. The index number of each name was selected according to the target line of each gate.

$$\begin{aligned}
&\text{1st gate } P[(1,2,3;1),(0,0,1)] : P_0 = (4,5)(12,13)(20,21)(28,29)\\
&\text{2nd gate } P[(3,4,5;3),(0,0,1)] : P_2 = (16,20)(17,21)(18,22)(19,23)\\
&\text{3rd gate } P[(1,2,5;5),(0,1,0)] : P_4 = (2,18)(6,22)(10,26)(14,30)\\
&\text{4th gate } P[(2,3,4;2),(0,0,1)] : P_1 = (8,10)(9,11)(24,26)(25,27)\\
&\text{5th gate } P[(1,2,3;1),(0,0,1)] : P_0 = (4,5)(12,13)(20,21)(28,29)\\
&\text{6th gate } P[(1,4,5;4),(1,0,0)] : P_3 = (1,9)(3,11)(5,13)(7,15)\\
&\text{7th gate } P[(1,2,3;1),(0,0,1)] : P_0 = (4,5)(12,13)(20,21)(28,29)
\end{aligned} \tag{6}$$

(2) Make the computation step First, we try to create the circuit by processing the third and fifth gates in parallel. The number of qubits common to both gates is 2, so at least two CWQs should be added to arrange these two gates in parallel. The indices of these qubits are 0 and 1, respectively. State values required for Toffoli gates in parallel

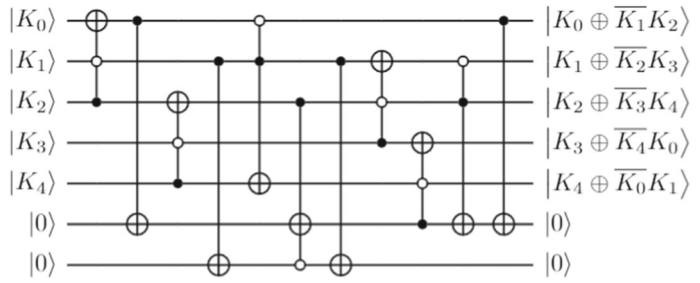

Fig. 5 χ quantum circuit with Toffoli-depth 6 and Toffoli-count 7 (χ-Z2). 4 CNOT gates seem to be used more than in χ-Z1 but it may be used more when considering decomposing Toffoli gates based on Clifford+T gate library

should be injected in CWQs before placing Toffoli gates parallelly. These state values can be injected by newly adding CNOT gates or changing the positions of existing Toffoli gates. In Fig. 5, the values in data qubits with index 0 and 1 are copied through CNOT gates to CWQs. The location of the CNOT gates installed in the front part of the circuit is arranged so that they are adjacent to Toffoli gates processed in parallel.

(3) Try to make the uncomputation step After the third and fifth gates are placed to be arranged in parallel, an initialization attempt is made for each CWQ. To be initialized, all inserted values in each CWQ should become zero by the XOR and AND operations. We first investigate a CWQ that corresponds to index 0. Of course, when calculating a state value in a CWQ, the gate using the CWQ as a control qubit can be ignored. Only gates using this qubit as a target qubit need to be considered when tracing the state value. In the case of this CWQ, this CWQ is initially injected with the state value $K_0 \oplus \overline{K_1}K_2$. Then, the state value $\overline{K_1}(K_2 \oplus \overline{K_3}K_4)(=(K_1 \oplus \overline{K_2}K_3)(K_2 \oplus \overline{K_3}K_4))$ is added. The state value generated by Toffoli gate and CNOT gate at the end of the circuit is the same as the state value $K_0 \oplus \overline{K_1}\overline{K_3}K_4$. The injected state values are duplicated. Therefore, the first CWQ can be initialized to the state $|0\rangle$. In this way, this CWQ contributes to Toffoli-depth reduction in the computation step and helps to generate other output values, and then is initialized to the state $|0\rangle$ using the state values remained in the uncomputation step (7).

Initialization for CWQ with index 0 in $\chi - Z2$ (Fig. 5):

$$(K_0 \oplus \overline{K_1}K_2) \oplus \overline{K_1}(K_2 \oplus \overline{K_3}K_4) \oplus (K_1 \oplus \overline{K_2}K_3)(K_2 \oplus \overline{K_3}K_4) \oplus (K_0 \oplus \overline{K_1}K_2) = 0 \tag{7}$$

This initialization is possible because two necessary conditions are satisfied. First, A Toffoli gate that shares the target line with the fifth MPT gate exists at the back of the circuit. Thanks to the existence of this gate, operations that cannot be made with NOT and XOR (Exclusive-OR) operations can be performed, such as AND operations in the uncomputation step for this CWQ. The target part of this seventh Toffoli gate P_0 is moved to a work qubit to initialize the work qubit. Second, the state values to be erased in work qubits should be created in the circuit beforehand. This condition is also satisfied because the function value $K_0 \oplus \overline{K_1}K_2$ was generated in the front part

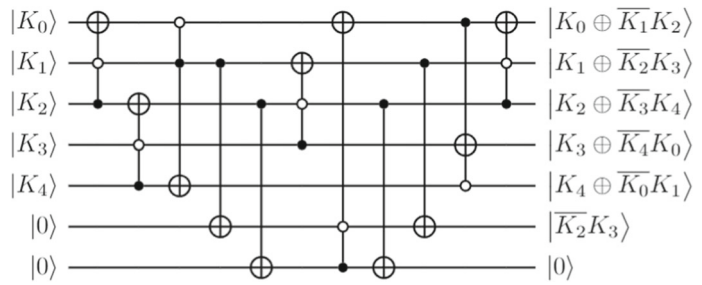

Fig. 6 χ quantum circuit with 2 work qubits with an incomplete uncomputation step. Unless Toffoli-count is increased, the state value K_1 of the first CWQ cannot be initialized with the remaining state values in the uncomputation step

of the circuit through the first MPT gate. (Each time a state value is injected, an XOR operation is performed, and we can check what state value is currently contained in the qubit.) Therefore, the uncomputation step for this work qubit can be well-constructed.

For the second CWQ, the initial state value K_1 in the qubit with index value 1 is injected through CNOT gate. Since there is no operation process for this index before the operation of the fifth Toffoli gate of χ-Z1 is performed, this CNOT gate only needs to be placed before the fifth Toffoli gate. This CWQ is used only as an off-control part of the parallelized Toffoli gate $P[(3, 6, 7; 6), (1, 0, 0)]$ in χ-Z2 circuit. Since the state value injected in the computation step is K_1 and this value exists in other qubits in the uncomputation step, the uncomputation step for this CWQ can be designed easily with only one CNOT gate.

(4) Try to reduce Toffoli-count (consequently Toffoli-depth) further As some Toffoli gates' control or target lines change, it's likely to reduce Toffoli-count (consequently Toffoli-depth) by the existing Toffoli-count reduction methods [22, 31–34]. Unfortunately, further Toffoli-depth reduction is not possible in the case of the χ-Z2 circuit. There are no Toffoli gates that share 2 control lines or 1 control line and 1 target line.

Another example: A failure case This time, let's determine whether the fourth and fifth MPT gates of χ-Z1 can be arranged in parallel (Fig. 6). As in the previous case, the number of commonly used qubits is 2, so it is required to add at least 2 CWQs. The common qubit indices are 1 and 2.

If the circuit is designed with the logic just explained, the uncomputation steps for one of two CWQs cannot be performed completely. This is because necessary conditions are not satisfied. There is no Toffoli gate with the same target line behind the fourth Toffoli gate P_1. Also, it is difficult to inject the state value K_1 back into the first CWQ without increasing Toffoli-count. The state value K_1 cannot be erased only by NOT and XOR operations.

Remark: Why cannot we use existing methods? Some readers may think that a circuit with reduced Toffoli-depth could be created by using existing methods, such as the Bidirectional and Simulated Annealing algorithms presented in 'Appendix A' after specifying the number of work qubits needed. However, since the Simulated Annealing algorithm uses DC as a cost function, it is difficult to return an exact circuit with an increased cost. The total design cost for the desired circuit will increase because the

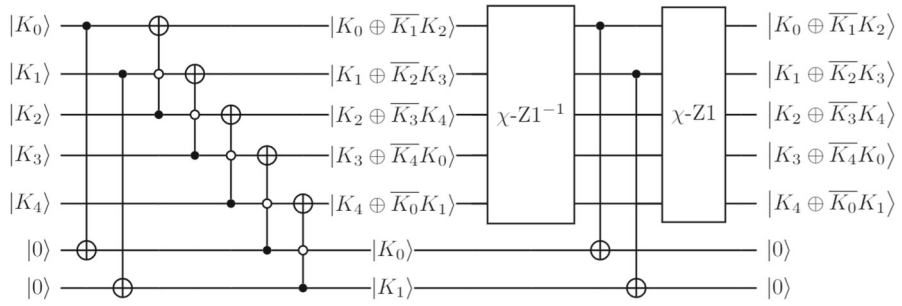

Fig. 7 χ quantum circuit with 2 CWQs. If we put this circuit as an input circuit to the simulated annealing algorithm, it is hard to get χ internal function circuit of Toffoli-depth 6 or less that we want. It probably returns the circuit χ-Z1 or the circuit with almost the same DC with χ-Z1

number of CNOT gates increases. That is, this Simulated Annealing algorithm cannot be used directly for our goal. As in Fig. 7, consider the circuit configuration assuming that two CWQs are determined to be used in advance. Two χ-Z1 circuits and two CNOT gates could be installed for the uncomputation step for these two CWQs. If this circuit is used as an input circuit for the simulated annealing algorithm, an output circuit that uses the amount of almost the same resource with χ-Z1 would be returned. Therefore, χ-Z2 with Toffoli-depth 6 or less is not likely to be created with the existing methods.

3.1.2 Case 2: Toffoli-count is allowed to increase

Sketch: Logic for the method's application to χ-Z1 We allow Toffoli-count to increase this time. When Toffoli-count increases are allowed, much more complex ideas are exploited. Therefore, the logic used when applying to the χ-Z1 circuit is briefly mentioned first through this Sketch subsection.

Some certain values are generated after passing through each gate in χ-Z1 circuit. Among these, there are not result values but intermediate state values that are created and then disappear. Intermediate values mean state values that are created and removed in the middle of a circuit and can be expressed as Boolean expressions like input and output state values. We paid attention to this intermediate state value's creation and destruction process. In the case of χ-Z1 circuit, the intermediate value $K_0 \oplus \overline{K_1 K_3} K_4$ is created in the middle part of the circuit, and this value is converted to $K_0 \oplus \overline{K_1} K_2$. Five of the seven Toffoli gates were involved in this generation and conversion process. The pair of result values of this five-gate circuit is ($K_0 \oplus \overline{K_1}K_2$, K_1, $K_2 \oplus \overline{K_3}K_4$, $K_3 \oplus \overline{K_4}K_0$, K_4). The states of the second and fifth qubits are maintained as K_1 and K_4, and the remaining three qubits consist of the result values we are looking for.

We wondered if there existed four other circuits that were symmetrical to this circuit and produced different function value pairs (8). ** stands for a 'don't care' output [1].

$$(**, K_1 \oplus \overline{K_2}K_3, **, K_3 \oplus \overline{K_4}K_0, K_4 \oplus \overline{K_0}K_1)$$
$$(K_0 \oplus \overline{K_1}K_2, **, K_2 \oplus \overline{K_3}K_4, **, K_4 \oplus \overline{K_0}K_1)$$

$$(K_0 \oplus \overline{K_1}K_2,\ K_1 \oplus \overline{K_2}K_3,\ **,\ K_3 \oplus \overline{K_4}K_0,\ **)$$
$$(**,\ K_1 \oplus \overline{K_2}K_3,\ K_2 \oplus \overline{K_3}K_4,\ **,\ K_4 \oplus \overline{K_0}K_1) \quad (8)$$

If such subcircuits exist, then we try to place a total of 5 subcircuits in parallel in one circuit. As a result, three χ operations could be made. That is, three pairs of overall result values ($K_0 \oplus \overline{K_1}K_2$, $K_1 \oplus \overline{K_2}K_3$, $K_2 \oplus \overline{K_3}K_4$, $K_3 \oplus \overline{K_4}K_0$, $K_4 \oplus \overline{K_0}K_1$) will be created, and two of these result value pairs can be initialized through CNOT gates located at the end of the circuit.

We were wondering how to find the subcircuits that produce the result values in expression (8). In particular, selecting the don't care outputs of these circuits was a major concern. Recall that our goal is to create a circuit with reduced Toffoli-depth. Therefore, in order to consider simultaneously finding four circuits that produce the result values in expression (8) while having a Toffoli-depth value of 7 or less, we decided to look for other forms of circuits that implement the complete χ function. In other words, we filled all the don't care outputs with the function values we were looking for. Then we took the necessary subcircuits from the χ-Z1 and other χ function circuits.

Since three χ operations are created, 10 work qubits (CWQs) should be added. The fact that three χ operations are made can be mathematically verified through permutation group theory before designing the resulting circuit. The permutation expression created through 25 gates used in 5 subcircuits is equivalent to exactly 3 χ operations (6,A2). Additionally, it may be possible to further reduce Toffoli-depth by removing and copying values through CNOT gates.

The method consists of 3 steps when Toffoli-count is allowed to increase like in the previous case. χ reversible function is used as a specific example again.

(1) Try to make initial circuits We first try to create circuits having the same Toffoli-count value with the input circuit χ-Z1. We call these circuits initial circuits. The permutation order in χ-Z1 circuit is in the following order (9).

$$P_0 \rightarrow P_2 \rightarrow P_4 \rightarrow P_1 \rightarrow P_0 \rightarrow P_3 \rightarrow P_0 \quad (9)$$

It tried to make an initial circuit where P_3 gate comes first among Toffoli gates. (As mentioned in Sect. 2, the control parts' activation conditions can be changed using NOT and CNOT gates, so it is enough to consider the placement of the P_3 gate only.) P_3 gate is moved to the beginning of the circuit. The current circuit is as follows (10). Two adjacent P_0 gates are canceled.

$$P_3 \rightarrow P_0 \rightarrow P_2 \rightarrow P_4 \rightarrow P_1 \quad (10)$$

Now, we add several MPMCT gates to convert back to an exact circuit. Conditional equations and positions of these gates can be found and expressed through permutation group theory. We compare the operation made by the current circuit and the operation made by χ internal function (A2,6) using permutation expressions. As a result, we can get an equation in permutation expressions (11).

$$(1, 9, 3, 11) = a_0(1, 9)(3, 11)a_1(9, 11)a_2 \quad (11)$$

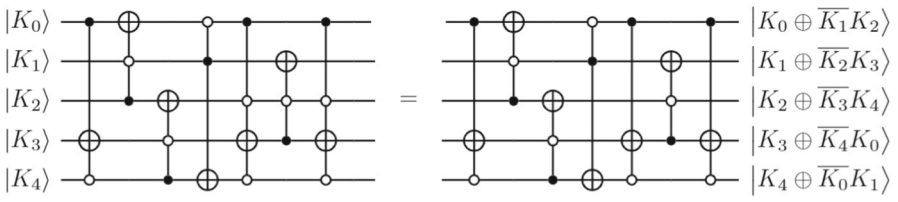

Fig. 8 Another version for χ quantum circuit without ancilla. Transformation rules were used while making this circuit

(1,9)(3,11) is obtained from P_3 and (9,11) is obtained from P_1 in the current circuit (10). It is not hard to get an answer a_0 = identity operation, and $a_1 = a_2$ = (1,9)(3,11). a_1 gate can be anywhere between the first P_3 and the last P_1 gate, and a_2 gate should be placed after P_1 gate. After installation, they can be transformed into two P_3 by local Toffoli-count reduction or transformation rules (Fig. 8) [22, 31–34]. Since a circuit with Toffoli-count 7 has been created, the simulated annealing algorithm does not need to be used as a global Toffoli-count reduction method in this case.

As mentioned earlier, we examine the intermediate value $K_0 \oplus \overline{K_1 K_3} K_4$ that are generated and disappear in the input circuit. There are a total of 5 gates used to generate and destroy the intermediate value, and the circuit made up of these gates has 3 of the 5 result values we want. Therefore, in order to complete the χ operation multiple times, we found five initial circuits including the input circuit (12).

Of course, it is possible to swap places between some gates in circuits without changing the entire operation by local transformation rules, and so some gates can be placed in a different order in the respective initial circuits. It was mentioned above that P_4 and P_0 and P_1 and P_0 are commutative in the input circuit χ-Z1.

$$\begin{aligned}
&P_0 \to P_2 \to P_4 \to P_1 \to P_0 \to P_3 \to P_0\\
&P_3 \to P_0 \to P_2 \to P_4 \to P_3 \to P_1 \to P_3\\
&P_1 \to P_3 \to P_0 \to P_2 \to P_1 \to P_4 \to P_1\\
&P_4 \to P_1 \to P_3 \to P_0 \to P_4 \to P_2 \to P_4\\
&P_2 \to P_4 \to P_1 \to P_3 \to P_2 \to P_0 \to P_2
\end{aligned} \tag{12}$$

(2) Try to implement a circuit that processes flows for intermediate values simultaneously After making the initial circuits, we proceed to the second step. We can estimate a lower bound for Toffoli-depth value of the output circuit by examining the process of generation and conversion for intermediate state values. To create the desired circuit, both the processes of generating intermediate state values in the initial circuits and converting them into final output values (or other intermediate state values that follow) should be included parallelly in the circuit. For example, when generating the intermediate value $K_0 \oplus \overline{K_1 K_3} K_4$ in χ-Z1 circuit, the necessary sequence of gates is P_0 and $P_2 \to P_0$. The sequences for gates needed to create other intermediate values are found from different initial circuits (13).

$$|0\rangle \to |K_0 \oplus \overline{K_1 K_3} K_4\rangle : P_0 \& P_2 \to P_0$$

$$
\begin{aligned}
&|0\rangle \rightarrow |K_1 \oplus \overline{K_2 K_4} K_0\rangle : P_1 \& P_3 \rightarrow P_1 \\
&|0\rangle \rightarrow |K_2 \oplus \overline{K_3 K_0} K_1\rangle : P_2 \& P_4 \rightarrow P_2 \\
&|0\rangle \rightarrow |K_3 \oplus \overline{K_4 K_1} K_2\rangle : P_3 \& P_0 \rightarrow P_3 \\
&|0\rangle \rightarrow |K_4 \oplus \overline{K_0 K_2} K_3\rangle : P_4 \& P_1 \rightarrow P_4
\end{aligned} \tag{13}
$$

Additional operations are required to convert all of these intermediate values into output values along with the initial values (14). Considering (13) and (14), it can be expected that Toffoli-depth of the output circuit will be at least 5. That means, the sum of Toffoli-depth for the (final) intermediate values generation and that for conversion of these intermediate values to the final output values is a lower bound for Toffoli-depth in the output circuit. Toffoli-depth could be reduced by the number of Toffoli gates that do not participate in the processes for generating and removing intermediate values in the initial circuits. As a side note, through the logic described above, an estimated value 25 for Toffoli-count could also be obtained.

$$
\begin{aligned}
&(|K_0 \oplus \overline{K_1 K_3} K_4\rangle, |K_3\rangle) \rightarrow (|K_0 \oplus \overline{K_1} K_2\rangle, |K_3 \oplus \overline{K_4} K_0\rangle) : P_3 \rightarrow P_0 \\
&(|K_1 \oplus \overline{K_2 K_4} K_0\rangle, |K_4\rangle) \rightarrow (|K_1 \oplus \overline{K_2} K_3\rangle, |K_4 \oplus \overline{K_0} K_1\rangle) : P_4 \rightarrow P_1 \\
&(|K_2 \oplus \overline{K_3 K_0} K_1\rangle, |K_0\rangle) \rightarrow (|K_2 \oplus \overline{K_3} K_4\rangle, |K_0 \oplus \overline{K_1} K_2\rangle) : P_0 \rightarrow P_2 \\
&(|K_3 \oplus \overline{K_4 K_1} K_2\rangle, |K_1\rangle) \rightarrow (|K_3 \oplus \overline{K_4} K_0\rangle, |K_1 \oplus \overline{K_2} K_3\rangle) : P_1 \rightarrow P_3 \\
&(|K_4 \oplus \overline{K_0 K_2} K_3\rangle, |K_2\rangle) \rightarrow (|K_4 \oplus \overline{K_0} K_1\rangle, |K_2 \oplus \overline{K_3} K_4\rangle) : P_2 \rightarrow P_4
\end{aligned} \tag{14}
$$

We can also roughly estimate how many CWQs will be needed. A circuit consisting only of gates involved in the intermediate value creation and deletion process returns only three result values in the desired output pair (8). Of course, the desired output pair is $(K_0 \oplus \overline{K_1} K_2, K_1 \oplus \overline{K_2} K_3, K_2 \oplus \overline{K_3} K_4, K_3 \oplus \overline{K_4} K_0, K_4 \oplus \overline{K_0} K_1)$, as shown in (A1). Therefore, if 5 subcircuits are processed in parallel, 3 output pairs will be created. Since 5 qubits are used to represent one output pair, a minimum of 10 CWQs are required. So about $(5+10)/5 = 3$ pairs of χ output values are expected to create in the desired circuit. A total of 15 qubits are expected to be used. CWQs are initialized using the output values and CNOT gates in the rear part of the output circuit.

(3) Check if Toffoli-depth could be reduced more The last step is to check if further Toffoli-depth reduction is possible. If the number of CWQs used when function values (or intermediate values) are created is twice more than the number of data qubits, then some qubits may be initialized easily and some values may be duplicated with only CNOT gates without Toffoli gates. Several qubits have the same input values (or intermediate values) at the middle part of the circuit made in the previous step. The number of CWQs used for the output circuit is 10, and after the first Toffoli-depth time slice, five input values (K_0 to K_4) exist duplicated, respectively. CNOT gates can be used to delete input values contained in five among ten work qubits and inject other existing values (15). It is noteworthy that CNOT gates can make the effect of Toffoli gates at this time. By using this last step, Toffoli-count and Toffoli-depth could be

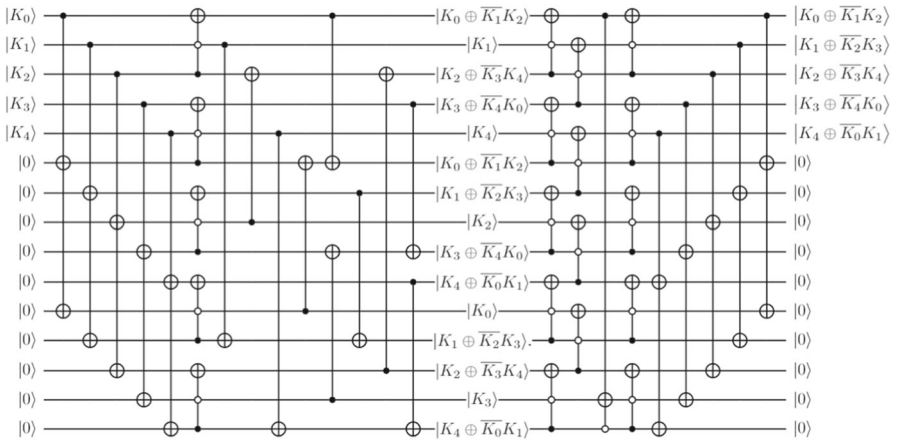

Fig. 9 χ quantum circuit with Toffoli-depth 4 and Toffoli-count 20 (χ-Z3). We can roughly estimate a lower bound for Toffoli-depth and Toffoli-count in advance. 20 Toffoli gates are grouped by 5 to form Toffoli-depth 4

reduced more. In conclusion, a circuit with Toffoli-count 20 and Toffoli-depth 4 was created (Fig. 9).

$$(K_0 \oplus \overline{K_1}K_2, K_0, K_0) \rightarrow (K_0 \oplus \overline{K_1}K_2, 0, K_0) \rightarrow (K_0 \oplus \overline{K_1}K_2, K_0 \oplus \overline{K_1}K_2, K_0) \tag{15}$$

Another example: A failure case It is not always possible to create in-place version circuits with reduced Toffoli-depth, even if we can build some initial circuits for a given input circuit. This is because processes in which intermediate values are created and removed in initial circuits should exist and also be able to be installed to process in parallel within the desired circuit. In Fig. 10, two initial circuits with Toffoli-count 2 are shown as an example. These circuits are expected to have the minimum Toffoli-count for output values. First of all, it is difficult to proceed with the second step because any intermediate value is not generated in the first initial circuit. Also, the number of duplicate values is not constant, so it is also hard to proceed with the last step. The number of work qubits is insufficient for initialization and value copying as well. It seems difficult to create a circuit with Toffoli-depth lower than the maximum value in multiplicative complexities among the function values. There may be future research about the relationship between Toffoli-depth and multiplicative complexities in the in-place circuit.

3.2 Our Toffoli-depth reduction method for in-place version circuits

In this subsection, the proposed method is described. Two sub-methods depend on whether Toffoli-count value increase is allowed. The input circuit for our method is assumed to be Toffoli-count optimized. NCT gate library is used during the process. Permutation group theory and Boolean algebra are utilized within the technique.

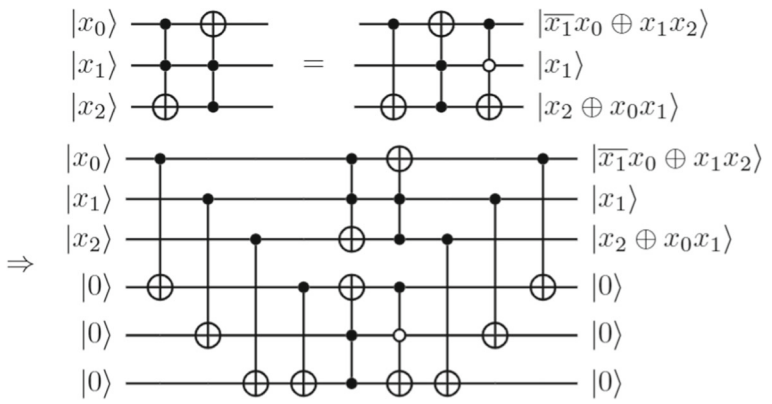

Fig. 10 A failure case when Toffoli-count value is allowed to increase. It seems hard to make Toffoli-depth smaller than the maximum value in multiplicative complexities among output values

3.2.1 Case 1: Toffoli-count value is not allowed to increase

When Toffoli-count increase is not allowed, the following procedure is performed.

1. *Try to find candidate pairs* For a given in-place version input circuit, find adjacent pairs of Toffoli gates that can be parallelized. That is, pairs of gates that are commutative with each other are tried to find. For example, if two gates share only control lines, they may be parallelized by adding CWQs. Commutativity for two Toffoli gates can be checked by their activation conditions or permutation group theory.
2. *Make the computation step* One of the gate pairs is selected and the computation step installation is attempted. CWQs are added as much as the number of lines (or the number of qubits) commonly used by gates in the pair. These two Toffoli gates are newly placed to achieve parallel processing. Of course, It is necessary to install CNOT gates between the data qubits and CWQs before placing these two Toffoli gates. These CNOT gates may be installed adjacent to the parallelized Toffoli gates.
3. *Try to make the uncomputation step* NOT and CNOT gates are placed so that the uncomputation step for CWQs is implemented. The desired reversible function (output) values are returned in the data qubits, up to swappings. The starting point of each CWQ's uncomputation step may be different.
4. *Try to reduce Toffoli-count (consequently Toffoli-depth) further* Finally, check if further Toffoli-count reduction is possible for the circuit created in the previous step. Even if Toffoli-count reduction is successful, Toffoli-depth may not be reduced.

As mentioned earlier, adding CWQs is essential, and the uncomputation step for these CWQs should be implemented in the output circuit. Finding a pair of parallelizable Toffoli gates does not always guarantee the generation for an in-place version quantum circuit with Toffoli-depth reduced. This is because CWQs may not be initialized and become garbage qubits. Therefore, two necessary conditions are required.

First, for at least one of the two Toffoli gates in a candidate pair, its target line and the target line of the third MPT gate behind them should be the same. Then, the uncomputation process for work qubits may be possible because only NOT and XOR (Exclusive-or) operations cannot produce an arbitrary binary operation like the AND operation. So when initializing CWQs, the target line position of some Toffoli gates may be changed.

Also, since the uncomputation step is completed through CNOT gates, the corresponding state values in work qubits should be created or remain in other qubits in advance. Each CWQ can be initialized only with state values remaining in data qubits and other CWQs after being used for parallel processing of Toffoli gates. CWQs are initialized if the values of all imported values are duplicated.

Due to these two conditions, the existence of a candidate pair does not always lead to the desired circuit. As a side note, the uncomputation step of each CWQ may start and end at different times. This may sound trivial since not all gates can perform operations simultaneously.

Some readers may wonder why this case is referred to as 'Toffoli-count is not allowed to increase.' Since an input circuit is assumed to have an optimized Toffoli-count value, it seems enough to say 'Toffoli-count is invariant.' However, as the number of work qubits increases and the positions of the control and target lines of the gates change (global) Toffoli-count reduction may become possible. Therefore, there is a further reduction step as the final step. Consequently, Toffoli-depth value may be further reduced according to the step. An example of a further reduction can be found in 'Appendix B.'

In addition to the methods mentioned in this main text, there is a case where Toffoli-depth is reduced obviously. This is a case where the existing state is restored in the middle of the circuit. In this case, Toffoli-depth can be easily reduced: The CWQ with the existing state can be initialized easily. Related examples can also be found in 'Appendix B.'

3.2.2 Case 2: Toffoli-count is allowed to increase

As Toffoli-count increases, DC of the result circuit tends to increase more than that of the given input circuit, so the existing DC optimization algorithms such as the simulated annealing algorithm cannot be used directly for Toffoli-depth reduction as mentioned above. The main idea is to try to arrange the processes for generating and removing intermediate values in different initial circuits parallel in one single circuit. There is room for Toffoli-depth to be reduced by the number of Toffoli gates, which are not related to the intermediate values. We investigate how many output values in the desired output pair are generated in each subcircuit consisting only of gates related to the generation and destruction of intermediate values. This allows us to know in advance how many subcircuits are needed and approximate Toffoli-depth value and number of CWQs of the resulting circuit. The process below is followed when Toffoli-count increase is allowed. The task of making several initial circuits with a given input circuit should be performed first.

1. *Try to make initial circuits* We attempt to create different versions of circuits with the same Toffoli-depth for the given input circuit. We look for circuits that implement

the same reversible function but have different orders of Toffoli gates. They are called initial circuits. When creating these circuits, we can get help from the existing synthesis and optimization algorithms. The reason for finding these initial circuits is to find subcircuits consisting of gates related to the intermediate values generated within each initial circuit. We call these subcircuits (or the sequence of these gates) essential circuits.

2. *Try to implement a circuit that processes flows for intermediate values simultaneously* It is tried to combine essential circuits (in initial circuits) made in the previous step into one circuit. Both the order of Toffoli gates used, and the process of creating and removing intermediate state values made in the initial circuits are investigated. That means the sequence for Toffoli gates can be gotten by checking the process of generating intermediate values and converting these values into the next intermediate values or output values. At the beginning of the result circuit, the initial state values of the data qubits are copied through CNOT gates. The number of CWQs needed can be estimated according to the number of output values in each essential circuit. If the size of the output value pair is n (which means that the bit-length of the function value is n), and the number of the output values that we look for in the essential circuit is m, then width of the final circuit may be lcm(n,m) (the least common multiple of n and m). Thus, the number of CWQs is the lcm value minus the bit-length n of the function value. (If the desired function is an odd permutation, the number of CWQs may be required to increase by the number of Toffoli gates parallel.) Gates are then installed in the circuit so that the processes related to intermediate values are processed in parallel. Toffoli-depth can be reduced by the number of Toffoli gates that do not participate in the generation and removal of intermediate values in the initial circuit. At the rear part of the circuit, all input and intermediate values disappear, so only output values remain as state values in qubits. Work qubits with common output values are initialized using CNOT gates.
3. *Check if Toffoli-depth could be reduced more* If the number of CWQs is 2n or more when the size of the domain of a given reversible function is 2^n, then Toffoli-depth and Toffoli-count may be reduced more. We check if Toffoli gates can be replaced with CNOT gates when duplicated state values exist.

In the first step, It is tried to see if several initial circuits can be made by changing the order of Toffoli gates used in the input circuit. These initial circuits may be made with the same Toffoli-count for the input circuit. One procedure is suggested to build one of these initial circuits. The algorithm used is the Simulated Annealing algorithm. It does not matter if other algorithms [33] are used when creating initial circuits instead of the simulated annealing algorithm because the global Toffoli-count optimization process is just required to create an initial circuit. An example of the process for the first step is as follows.

1. We choose one Toffoli gate in a given input circuit and move to it at the front of the circuit.
2. Several MPMCT gates are installed so the desired (reversible) function operations are restored in the current circuit. MPMCT gates to be installed and their positions can be found through permutation group theory. After installation, this exact circuit may have an increased Toffoli-count value. Toffoli-count may be reduced through

local Toffoli-count reduction or transformation rules [22, 31–34]. If a circuit with the same Toffoli-count for the input circuit is made, there is no need to go to the next step.

3. The simulated annealing algorithm can be used as a global Toffoli-count reduction method in this step. This algorithm takes the exact circuit made in the previous step as an input circuit. It is attempted to make a circuit with the same Toffoli-count, while the order of Toffoli gates is different. If any circuit desired is not made up to this step, it means that it is difficult to make the initial circuit have the same Toffoli-count value as the input circuit.

In fact, the process of creating this initial circuit can be made very simple in special cases. If the output values have all the same form, simply relabeling and rearranging the qubit values are sufficient [35]. The above method describes the process of creating an initial circuit in general.

Before starting the second step, the approximate number of CWQs required can be estimated by checking the number of output values in the essential circuits. Also, Toffoli-depth of the output circuit can be estimated by examining the creation and destruction processes for intermediate values in advance. Toffoli-depth could be reduced by the number of Toffoli gates that do not participate in the process of generating and transforming intermediate values in the initial circuits.

4 Quantum resources comparison

We consider χ internal function block in SHA3-256 as a concrete example in our proposed method. Five versions of χ internal function circuits are presented. We additionally describe designing the fourth and fifth version quantum circuits using MBQC in 'Appendix C.' We compare the quantum resources required in χ circuits made in previous studies with those of the circuits made by our method. Five versions of the entire SHA3-256 circuit can be proposed according to our χ circuits' version, and their quantum resources are also compared with those of the circuits in previous studies [3, 6, 18]. A metric Width/T-depth is used to show that our circuits are well-balanced.[2] Lastly, we created Grover's algorithm circuits based on the SHA3-256 circuits and calculated quantum resources for these circuits. By multiplying Width and Toffoli-depth values, it is confirmed that the algorithm circuit is designed more efficiently when the circuit applying our technique is used than when it is not.

[2] Some readers might think this metric is inappropriate for explaining the 'balance' of the quantum circuit and the term 'balance' doesn't need to be mentioned. We want to show the difference between in-place and out-of-place circuits numerically.

4.1 Quantum resources comparison for χ internal function

Table 1 shows the quantum resources required when designing each version circuit and those for the circuits in previous studies [3, 6, 18]. Since these χ circuits are based on the NCT gate library, T-depth is optimized using T-depth reduction techniques after decomposing Toffoli gates into Clifford+T gates [8–10].

All of the proposed circuits are in-place version quantum circuits that do not require an inverse function circuit unlike the circuit in the previous study [18]. χ-Z1 has fewer qubits and Toffoli-depth (and T-depth) than $\chi+\chi^{-1}$ circuit [18]. Since χ-Z2 uses more qubits and T and $T^{\dagger}$ gates than χ-Z1, DC is bigger at the FTQC level as well as the logical level. However, since T-depth is smaller than that of χ-Z1, the execution time is expected to be shorter when considering FTQC. The optimized T-depth for χ-Z3 is 8, so it seems not much different from the performance of χ-Z2. χ-Z4 and χ-Z5 are circuits made using MBQC ('Appendix C'). The execution time may be shorter or longer than χ-Z2 or χ-Z3, respectively, depending on which physical system is used to design the quantum circuit. In the case of χ-Z5 circuit, Toffoli-count 20* refers to the sum of the number of AND and $AND^{\dagger}$ gates with the number of Toffoli gates. Similarly, Toffoli-depth 4* includes values created by AND or $AND^{\dagger}$ gates. AND-depth and AND-count occupy half of Toffoli-count and Toffoli-depth each.

Width/T-depth values were not written in this table because Width values of SHA3-256 quantum circuits in the previous papers can be reduced [3, 6]. The number 20 for Width in the table can be checked in 'Appendix D.' Previous papers did not describe their χ circuits in detail.

4.2 Quantum resources comparison for SHA3-256 quantum circuit

Before calculating quantum resources for SHA3-256 circuit, the number of input message blocks should be determined first. It is assumed that the number of blocks in a padded message is just one which means that the length of the original message is less than 1087 bits. A more detailed explanation can be found in [36].

Five SHA3-256 quantum circuits are created respectively by combining one of five χ quantum circuits with the other internal function quantum circuits. In other words, χ quantum circuits used in each version are different, and the quantum circuits that implement the remaining four internal functions are all the same. These four internal function blocks can be implemented as an ancilla-free circuit with only CNOT and NOT gates without Toffoli gates [18]. As a side note, θ function can be designed through PLU decomposition or existing linear reversible synthesis methods without providing work qubits because it is a linear reversible transformation [37–39]. The remaining three functions (ρ, π, and ι) also do not require the help of any work qubits, as mentioned in the previous study [18].

In Keccak-f, one of the internal functions of SHA3-256, 320 5-bit χ internal functions are processed in parallel over 24 rounds. That is, χ internal function is used 7680 times in one Keccak-f. Keccak-f quantum circuits made in other studies use work qubits excessively a lot because they are out-of-place version circuits [3, 6]. (In fact, their Width values (46,400 and 44,798) can be reduced to 43,200 without any change

Table 1 Quantum resources for different χ quantum circuits

	Width	#ancilla	T-depth	T-count	Toffoli-depth	Toffoli-count	Remark
$\chi+\chi^{-1}$ [18]	10	5	33 → 18	77 → 65	11	11	In-place
χ [3, 6]	20	15	1	20	1*	5*	Out-of-place
χ-Z1	5	0	21 → 11	49 → 37	7	7	In-place and input
χ-Z2	7	2	18 → 9	49 → 43	6	7	In-place and output
χ-Z3	15	10	12 → 8	140 → 110	4	20	In-place and output
χ-Z4	7	2	15 → 8	39 → 35	6*	7*	In-place and MBQC
χ-Z5	15	10	8 → 6	90 → 80	4*	20*	In-place and MBQC

The change of T-depth and T-count is occurred by T-depth (and T-count) reduction methods [8–10]. Toffoli-depth and Toffoli-count for χ-Z4 and Z5 are numbers including AND-depth and AND-count, respectively

for T-depth and T-count by using CNOT gates for initializing qubits as mentioned above. See 'Appendix D' for more detail.) When T-depth is more considered than Depth, an out-of-place version circuit for a cryptosystem may look like an inappropriate circuit for use in Grover's algorithm. This is because when an MPMCT gate existing in the Oracle operator is attempted to be decomposed into Toffoli gates, it can be decomposed more efficiently with the help of CWQs rather than DBQs so T-depth of the entire circuit could be smaller [21, 40, 41]. A well-balanced circuit configuration for Grover's algorithm may be achieved using an in-place version subcircuit for the cryptosystem. The circuit for Grover's algorithm is examined in the next subsection.

Each circuits made are named SHA3-256-Z1, SHA3-256-Z2, SHA3-256-Z3, SHA3-256-Z4, and SHA3-256-Z5, respectively (Table 2). For SHA3-256-Z1, both width and T-count are smaller than the previous SHA3-256 circuit [18], so DC is lower. Also, since T-depth is smaller, the overall operation is performed in a shorter execution time. SHA3-256-Z2 is also an upward compatible circuit of the previous SHA3-256 circuit [18], and there is a trade-off between the quantum resources of SHA3-256-Z1 and those of SHA3-256-Z2. SHA3-256-Z4 is a circuit in which MBQC is applied to the SHA3-256-Z2 circuit, and some Toffoli gates are changed to AND or $\text{AND}^{\dagger}$ gates, so T-count and T-depth are reduced. Width of SHA3-256-Z3 and Z5 is about 11% of that of the previous Keccak-f quantum circuit [6]. As mentioned above, the superiority or inferiority of the two circuits (SHA3-256-Z3 and Z5) depends on which physical system they are based on, that is, according to the performance of (intermediate) measurement meters in a given quantum environment. The performance difference between SHA3-256-Z2 and SHA-256-Z4 is also determined for the same reason.

The suggested circuits' quantum resources are more well-balanced than those required for designing previous SHA3-256 quantum circuits. A metric called Width/T-depth is used to show the balances of the circuits. For previous out-of-place version circuits, Width/T-depth values are all greater than 1866. On the other hand, all the circuits we made had both width/T-depth values less than 34, which is less than about 1/55 times the values of the previous out-of-place version circuits.

4.3 Quantum resources comparison for Grover's algorithm

Some readers may be unsure about the usability of the presented method. One could ask whether it is simply a trade-off between time complexity and space complexity because width increases and Toffoli-depth (and consequently T-depth) decreases.

We designed the entire Grover's algorithm using the SHA3-256 circuit we created. When measuring the security strength of a cryptographic system (symmetric key cryptosystem, hash algorithm, etc.) in a quantum environment, it is not to calculate the quantum resources needed to design the cryptosystem but to ultimately calculate the quantum resources needed to design Grover's algorithm.

We created several versions of the algorithm circuit and calculated the quantum volume for each circuit. In this paper, quantum volume refers to Toffoli-depth and Width of a quantum circuit multiplied by a ratio of 1:1 (Table 3). Of course, this 1:1 ratio is a value we arbitrarily set and may vary depending on the future direction of quantum computer development. (If the decoherence constraint is relaxed, the proportion of

Table 2 SHA3-256 quantum circuit resources comparison

	Width	Width/T-depth	T-depth	T-count	Toffoli-depth	Toffoli-count	Remark
SHA3-256 [18]	3200	7.41	432	499,200	264	84,480	In-place
Keccak-f [3]	46,400	1933.33	24	153,600	24*	38,400*	Out-of-place
Keccak-f [6]	44,798	1866.58	24	153,600	24*	38,400*	Out-of-place
SHA3-256-Z0	43,200	1800	24	153,600	24*	38,400*	Out-of-place
SHA3-256-Z1	1600	6.06	264	284,160	168	53,760	In-place and input
SHA3-256-Z2	2240	10.37	216	330,240	144	53,760	In-place and output
SHA3-256-Z3	4800	25	192	844,800	96	153,600	In-place and output
SHA3-256-Z4	2240	11.67	192	268,800	144*	53,760*	In-place and MBQC
SHA3-256-Z5	4800	33.33	144	614,400	96*	153,600*	In-place and MBQC

Five versions of SHA3-256 quantum circuit having the in-place property are made. Since they are all in-place version quantum circuits, they provide CWQs rather than DBQs to the comparator (the MPMCT gate) in Grover's algorithm. These are well-balanced circuits when considering both space and time complexity. SHA3-256-Z1 and Z2 are superior to the in-place version quantum circuit in the previous research [18]. SHA3-256-Z3 can provide 3200 CWQs, not DBQs for the next operation design in a circuit. For SHA3-256-Z4 and Z5, Toffoli-count and Toffoli-depth values include AND-count and AND-depth values, respectively. A description of SHA3-256-Z0 can be found in 'Appendix D'

Table 3 Quantum resources for Grover's algorithm where the cryptosystem is SHA3-256 and the length $|M|$ of the message is 1086

SHA3-256	Width	Toffoli-depth for SHA3-256	Toffoli-depth for Grover's algorithim	Width × Toffoli-depth for Grover's algorithim
SHA3-256 [18]	3200	264	$1.5202\ldots \times 2^{136}$	$1.1876\ldots \times 2^{148}$
Keccak-f [3]	46400	24	$1.8113\ldots \times 2^{133}$	$1.2824\ldots \times 2^{149}$
Keccak-f [6]	44798	24	$1.8113\ldots \times 2^{133}$	$1.2381\ldots \times 2^{149}$
SHA3-256-Z0	43200	24	$1.8113\ldots \times 2^{133}$	$1.1939\ldots \times 2^{149}$
SHA3-256-Z1	1600	168	$1.8692\ldots \times 2^{137}$	$1.4603\ldots \times 2^{148}$
SHA3-256-Z2	2240	144	$1.7466\ldots \times 2^{135}$	$1.9103\ldots \times 2^{146}$
SHA3-256-Z3	4800	96	$1.2291\ldots \times 2^{135}$	$1.4403\ldots \times 2^{147}$

Grover's algorithm circuits were designed for each SHA3-256 circuit created, and quantum volume (Width × Toffoli-depth) values were calculated for comparison. It can be seen that the two cases (SHA3-256-Z2 and Z3) where the circuit to which our method is applied is used are all more efficient than the case (SHA3-256-Z1) where the method is not

Width will decrease, and if the noise constraint is relaxed, the proportion of Toffoli-depth will decrease.) Grover iteration number followed the results of one previous study [15]. In the study, the optimal number of Grover iterations was investigated, and the number is $0.690\ldots \cdot \sqrt{2^{256}}$ for SHA3-256. A single Grover iteration consists of two cryptosystem circuits and two different MPMCT gates. How to efficiently design MPMCT gates based on T-depth can be checked in our previous research [41]. The length of the message was set to 1086, which is the maximum length that one message block can contain as mentioned above. (As a side note, there is no need to add any work qubits when creating Grover's algorithm circuit [42]. The entire algorithm circuit can be created from the number of qubits utilized to make the cryptosystem circuit.) Since the length of the hash value is 256 bits, C^{255}NOT gate can be installed in the oracle operator, and since the length of the original message is 1086 bits, C^{1085}NOT gate can be installed in the diffusion operator.

From the table, first of all, it can be seen that Grover's algorithm circuit can be designed more efficiently when the cryptosystem is created as an in-place version. (SHA3-256 in [18], SHA3-256-Z1, Z2, and Z3.) This is because CWQs can be provided to MPMCT gates, which help the MPMCT gates to be designed efficiently.

To understand the usefulness of the Toffoli-depth reduction technique we proposed, it is reasonable to compare the case where the SHA3-256-Z1 circuit is used and the case where SHA3-256-Z2 and Z3 are used. By using the SHA3-256 quantum circuits created through the presented method as subcircuits, Grover's algorithm circuit can be designed much more efficiently. If Toffoli-count increase is not allowed, quantum volume value becomes approximately 1/3 of the existing value. If Toffoli-count increase is allowed, quantum volume value becomes about half of the existing value.

5 Conclusion

5.1 Discussion

A methodology is proposed to return an in-place version circuit with reduced Toffoli-depth given an in-place version circuit with Toffoli-count optimized as an input circuit. Our proposed method is considered divided into two cases. Based on Toffoli-count value of the input circuit, one case does not allow an increase, and the other case allows it. In the former case, candidate pairs of Toffoli gates that may be arranged in parallel are tried to find, and then the number of required CWQs could be roughly identified. In the latter case, the core idea is that several different sequences of Toffoli gates with generating and conversion for intermediate state values are tried to process in parallel in one circuit. In both cases, additional use for CWQs is inevitable, but CWQs are not excessively added like in previous studies that made out-of-place version circuits [3, 6].

As mentioned earlier, the two main constraints preventing the development of quantum computers are decoherence and noise problems. Currently, it is unknown which restriction will be relaxed in the future, so creating various versions of circuits that realize the same operation is a reasonable choice. T-depth (and T-count) might be more

critical than Width and FTQC will be considered more in the future so Toffoli-depth reduction process should be included before processing T-depth optimization process. Accordingly, multiple versions of circuits can be created through the proposed Toffoli-depth reduction technique, and this method can ultimately be used as a tool to more efficiently design application circuits such as Grover's algorithm.

Many research projects have been and are still being done to optimize the circuits of cryptographic systems such as symmetric key cryptosystems and hash algorithms. These circuits tend to enter as subcircuits within Grover's algorithm. Therefore, a more reasonable approach is to ultimately aim to optimize Grover's algorithm circuit rather than optimizing these subcircuits. When optimizing any cryptosystem circuit used in Grover's algorithm, we should consider how to provide sufficient CWQs to MPMCT gates, which is another subcircuit.

χ internal function block in SHA3-256 was used to explain as an example. Five different versions of χ circuits are presented. Five versions of SHA3-256 quantum circuits depend on which χ circuit version is used, and these circuits are all well-balanced in-place version circuits with appropriate width and Toffoli-depth values. A measure width/T-depth is introduced to show that the presented circuits are well-balanced. Each circuit made is suitable for use as an internal block in Grover's algorithm because CWQs rather than DBQs can be provided to the MPMCT gate decomposition, which is the next operation. It was experimentally confirmed that Grover's algorithm circuit was designed more efficiently when using the SHA3-256 circuits to which our method was applied. As a side note, ideas needed when utilizing MBQC are also presented in 'Appendix C.'

5.2 Future work

This Toffoli-depth reduction technique preserving the in-place property for Toffoli-count-optimized circuits is presented for the first time to the best of our knowledge, so it is not easy to find cases where the technique is applied successfully. Additional research is needed to see if there is a circuit class to which this technique can be applied. In particular, it is not easy to find initial circuits that perform the same (reversible) operation and have the same Toffoli-count value. If such circuit classes exist and that circuit can be found beforehand, meaningless work for the next step attempts could be avoided. Although all the output values have the same form, our method does not always seem to be applicable.

When increasing Toffoli-depth is allowed, initial circuits with the same Toffoli-count for a given input circuit are tried to create. In making these circuits, one of the transposition solutions should be found to generate the original (reversible) function, but finding this transposition solution does not seem always easy. The time complexity of this search step for making initial circuits may be studied in the future.

Any method that can prove mathematically that Toffoli-depth or Toffoli-count is optimized has not been suggested yet. This problem of getting optimal Toffoli-count (Toffoli-depth) seems to relate to the minimum circuit size problem. The MCS problem has not been solved in theoretical computer science for decades. If a mathematical theory can be found or developed that explains whether Toffoli-depth and Toffoli-count

are optimized, it would be the counterpart of the T-depth and T-count optimization technique [8, 43]. This theory will explain the trade-off between Toffoli-count and Toffoli-depth, and determine the number of work qubits required.

Appendix A: Existing reversible synthesis and optimization methods for the input circuit in our proposed method

In this section, several existing techniques are introduced that can be used in generating the input circuits of our presented technique. We first introduce a reversible synthesis method that produces accurate circuits called the bidirectional algorithm. The output circuit from this algorithm can be an input circuit for an optimization technique. Among the optimization techniques, the simulated annealing algorithm is used that employs DC as a cost metric. The output circuit from this optimization technique could be an input circuit for our method. When our technique is applied, an in-place version circuit with Toffoli-count optimized and based on the NCT gate set is used as the input circuit. After making the input circuit for our proposed method, the reason that the simulated annealing algorithm can be used as a global Toffoli-count reduction technique is mentioned briefly. A proper input circuit for our method may be created through different synthesis and optimization methods that were proposed before but not mentioned here [33, 44].

A.1 Synthesis method: bidirectional algorithm

An accurate circuit generation is attempted for a given truth table or (reversible) function. Since the given reversible function is a one-to-one correspondence function, it can be expressed as a permutation. It is known that even permutation can be implemented as a reversible circuit without work qubits [17]. Meanwhile, if a given function is an odd permutation, an in-place version reversible circuit based on the NCT gate set can be created only with the help of at least one work qubit. Therefore, it is possible to determine in advance how many work qubits are required before making an accurate in-place version circuit from a synthesis algorithm.

In a previous study, a reversible circuit synthesis method called the Bidirectional algorithm using a given truth table was presented [22]. The reason this is called the Bidirectional algorithm is that it considers the placement of gates at the front and back sides of the quantum circuit. The gates placed when implementing a circuit using this algorithm are MPMCT gates. Locations and gates are selected and placed repeatedly while the algorithm is running. When the number of Boolean variables considered for a truth table or invertible function is n, the simplest gate available is the NOT gate and the largest gate is the C^{n-1}NOT (or MP-(n-1)-controlled-NOT) gate. Circuits based on MPMCT gates can be converted into circuits based on NCT gates with the help of work qubits [45].

As an example for this algorithm, χ internal function in SHA3-256 is used. χ internal function is represented by the expression (A1) in Boolean algebra [36]. For

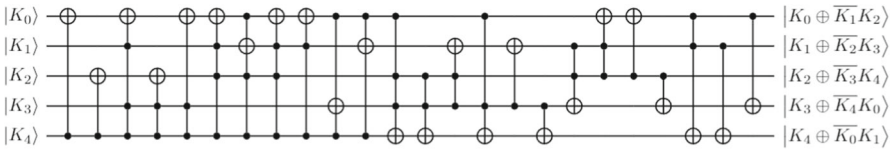

Fig. 11 χ quantum circuit construction with the bidirectional algorithm. The circuit in the figure consists only of CNOT, C^2NOT, and C^3NOT gates. According to a previous study, C^3NOT gate can be decomposed into four C^2NOT gates [45]. Therefore, this circuit can be composed only of NCT gates

$X \in \{0,1\}, \overline{X} = X \oplus 1 = 1 - X.$

$$(K_4, K_3, K_2, K_1, K_0) \Rightarrow (K_4 \oplus \overline{K_0}K_1, K_3 \oplus \overline{K_4}K_0, K_2 \oplus \overline{K_3}K_4, K_1 \oplus \overline{K_2}K_3, K_0 \oplus \overline{K_1}K_2) \tag{A1}$$

Table 4 shows the truth table for χ internal function. From the truth table, it can be seen that this function is a 5-variable reversible transformation and is represented by 5 transpositions and 5 4-cycles (A2).

$$\text{The permutation for } \chi \text{ internal function block: } (0)(1, 9, 3, 11)(2, 18, 6, 22)(4, 5, 12, 13)(7, 15)(8, 10, 24, 26)(14, 30)(16, 20, 17, 21)(19, 23)(25, 27)(28, 29)(31) \tag{A2}$$

Since a cycle of length 4 can be decomposed to 3 transpositions, χ internal function can be expressed as a product of 20 transpositions. It is an even transposition, so it can be implemented as an NCT gate-based ancilla-free circuit. The quantum circuit in Fig. 11 is made through the Bidirectional algorithm.

Looking at this figure, any C^4NOT gate, which uses five qubits, is not needed to implement χ internal function as a reversible circuit. Also, since the C^3NOT gate can be expressed as four Toffoli gates, it can be considered that 31 Toffoli gates were used in this circuit [45].

A.2 Optimization method: simulated Annealing algorithm

The optimization process is usually performed after the correct circuit has been created. Since our proposed method is for Toffoli-depth reduction, it is meaningful to apply the technique only when an input circuit with Toffoli-count optimized is used. If Toffoli-count is minimal optimized, then Toffoli-depth could not be reduced more just by (local) Toffoli-count reduction rules without increasing Width. Therefore, the cost metric we want at this time is Toffoli-count. There have been studies that have attempted to lower DC values by suggesting a local Toffoli-count reduction (optimization) method based on template matching or transformation rules [33, 34]. Unfortunately, these methods are not global Toffoli-count reduction techniques and do not guarantee the return for the optimal circuit. As far as we know, many local Toffoli-count reduction methods have been proposed, but no method is suggested to get the optimal Toffoli-count (or Toffoli-depth) provably or efficiently [33].

Table 4 Truth table of χ internal function block

Input	Decimal representation	Output	Decimal representation	Input	Decimal representation	Output	Decimal representation
00000	0	00000	0	10000	16	10100	20
00001	1	01001	9	10001	17	10101	21
00010	2	10010	18	10010	18	00110	6
00011	3	01011	11	10011	19	10111	23
00100	4	00101	5	10100	20	10001	17
00101	5	01100	12	10101	21	10000	16
00110	6	10110	22	10110	22	00010	2
00111	7	01111	15	10111	23	10011	19
01000	8	01010	10	11000	24	11010	26
01001	9	00011	3	11001	25	11011	27
01010	10	11000	24	11010	26	01000	8
01011	11	00001	1	11011	27	11001	25
01100	12	01101	13	11100	28	11101	29
01101	13	00100	4	11101	29	11100	28
01110	14	11110	30	11110	30	01110	14
01111	15	00111	7	11111	31	11111	31

It is a 5-variable reversible function and of course, can be expressed as a permutation. If we express it as a product of transpositions, it can be seen that it is an even permutation (A2)

We selected the simulated annealing algorithm [1] to reduce Toffoli-count when an input circuit for our method is made. (As a side note, width is invariant while using this algorithm.) This machine-learning-based heuristic method lowers DC. In fact, the input circuit for this algorithm may be set to any circuit, but the exact circuit is recommended to be used so that the returned circuit is also likely to be exact. In addition, an arbitrary set of gates can be used, and DC for each gate should be determined in advance. When using this algorithm, the selected gate set consists of NOT, CNOT, Toffoli, and MPT gates. Each DC is set to 1, 1, 5, and 5 according to the logic in [1]. Since the cost for Toffoli gates is much higher than NOT and CNOT gates, this DC reduction method can be used as a global Toffoli-count reduction method for an NCT-gate-based circuit. For each iteration in the algorithm, the duration decreases as the temperature goes from high to low, and a gate within a preselected set and a location in the quantum circuit are selected. Then, one of the add, delete, and replace operations is performed. This work creates a neighboring circuit, and its DC is compared with the current circuit's. Then, the circuit with the lower DC value is selected. At first glance, this algorithm may look for a circuit with a local minimum DC value, but because random elements are used, the globally minimized solution could be returned. Also since this algorithm is probabilistic, it may return an incorrect circuit. That is, it may return a near-correct circuit, not a correct circuit for the desired reversible function. Therefore, several repetitive works might be needed to return an accurate circuit with DC reduced. When an accurate circuit with reduced DC was returned, this returned output circuit was used as the input circuit again for this Simulated Annealing algorithm. We discard the output circuit when it is not exact.

As a result of performing about dozens of times, an in-place version χ internal function quantum circuit consisting of only 7 MPT gates was obtained. We named this circuit χ-Z1 (Fig. 4). It is an ancilla-free circuit in which no work qubit is used at all. This circuit is used in our proposed method as an input circuit.

The simulated annealing algorithm can be used as a global Toffoli-count reduction technique because DC value for Toffoli gates can be set as significantly higher than those for other gates. (For example, DC values for Toffoli or MPT gates could still be set to 5 but DC values for other gates could be set to 0.) Also, since the gate set can be determined in advance, it is suitable for making an in-place version circuit based on the NCT gate library.

Compared to a circuit created previously (Fig. 12, [18]), it can be seen that both width and Toffoli-depth (and Toffoli-count) are reduced. They made a χ^{-1} circuit separately to configure the uncomputation step. They designed this step by exchanging the roles of data qubits and work qubits so it is implemented slightly differently from the usual way. Looking at χ and χ^{-1} function circuits, data qubits ($(|K_t\rangle$ $(t = 0, \ldots, 4))$) are changed to CWQs ($|0\rangle$). This circuit overuses various quantum resources such as qubits and gates. On the other hand, it is expected that our ancilla-free χ-Z1 circuit is Toffoli-count optimized among in-place version χ circuits (Fig. 4).

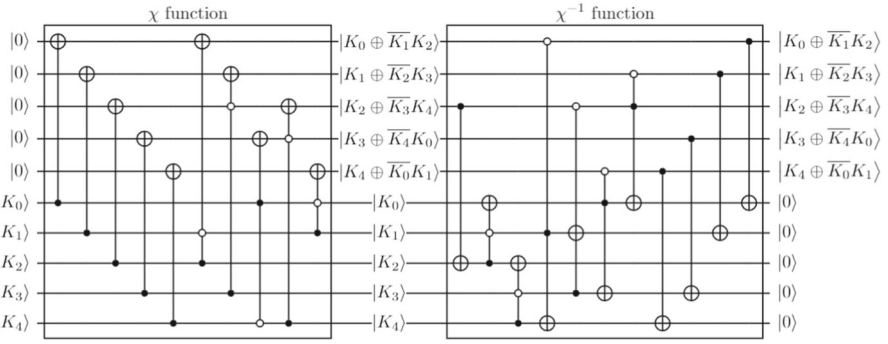

Fig. 12 χ and χ^{-1} quantum circuits in a previous work [18]. In fact, χ and χ^{-1} function blocks were not presented in detail in the previous study as the figure. χ^{-1} function circuit shown in the figure are made ourselves

Appendix B: Examples for our presented method

We present three successful examples of applying the proposed method here. It was very difficult to find successful cases using existing benchmarks [35]. Therefore, we created these examples ourselves. The created reversible functions can be expressed as Boolean expressions as follows (B3). They are called f_1, f_2, and f_3, respectively.

$$
\begin{aligned}
&f_1 : (K_6, K_5, K_4, K_3, K_2, K_1, K_0) \Rightarrow \\
&(K_6 \oplus \overline{K_0}K_1,\ K_5 \oplus \overline{K_6}K_0,\ K_4 \oplus \overline{K_5}K_6,\ K_3 \oplus \overline{K_4}K_5,\ K_2 \oplus \overline{K_3}K_4,\ K_1 \oplus \overline{K_2}K_3, \\
&K_0 \oplus \overline{K_1}K_2) \\
&f_2 : (K_4, K_3, K_2, K_1, K_0) \Rightarrow \\
&K_4 \oplus \overline{K_3}K_2 \oplus \overline{K_3}\,\overline{K_1}K_0,\ K_3 \oplus \overline{K_2}K_1 \oplus \overline{K_2K_0}K_4,\ K_2 \oplus \overline{K_1}K_0 \oplus \overline{K_1K_4}K_3, \\
&K_1 \oplus \overline{K_0}K_4 \oplus \overline{K_0}\,\overline{K_3}K_2,\ K_0 \oplus \overline{K_4}K_3 \oplus \overline{K_4K_2}K_1) \\
&f_3 : (K_4, K_3, K_2, K_1, K_0) \Rightarrow \\
&(K_4 \oplus \overline{K_3}\,\overline{K_1}K_0,\ K_3 \oplus \overline{K_2K_0}K_4,\ K_2 \oplus \overline{K_1K_4}K_3,\ K_1 \oplus \overline{K_0K_3}K_2,\ K_0 \oplus \overline{K_4K_2}K_1)
\end{aligned}
\tag{B3}
$$

f_1 is a 7-bit reversible function, and f_2 and f_3 both are 5-bit reversible functions (Table 5). Among the circuits that implement each function, all input circuits are assumed to be Toffoli-count optimized circuits. Since they are all even permutations, input circuits can be implemented without using any work qubits. As a side note, there is previous research on determining whether a given Boolean function is reversible [46].

B.1 Funtion f_1

f_1 has output state values of the same form as the χ function. That's why the form of the input circuit f_1-Z1 is similar to the form χ-Z1 (Fig. 13). Two pairs of Toffoli gates can operate parallel without any work qubits, so Toffoli-depth value of f_1-Z1 is 9. Therefore, it is meaningless to apply the method when an increase in Toffoli-count value is not allowed because Toffoli-depth no longer decreases. However, we

Table 5 Examples for our presented method

	Width	#ancilla	Toffoli-depth	Toffoli-count	Remark
f_1-Z1	7	0	9	11	In-place and input
f_1-Z2	9	2	9	11	In-place and output
f_1-Z3	28	21	6	42	In-place and output
f_2-Z1	5	0	7	7	In-place and input
f_2-Z2	7	2	6	7	In-place and output
f_2-Z3	15	10	4	20	In-place and output
f_3-Z1	5	0	12	12	In-place and input
f_3-Z2	6	1	11	12	In-place and output
f_3-Z3	6	1	11	11	In-place and output
f_3-Z4	15	10	5	25	In-place and output

A total of 10 circuit figures for 3 different functions are presented in this 'Appendix'

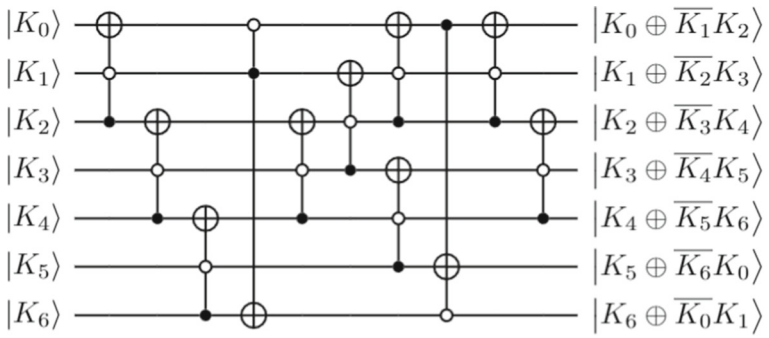

Fig. 13 f_1-Z1 circuit. Toffoli-count value is 11 and Toffoli-depth value is 9

drew a figure to show an example circuit where the proposed method was applied successfully (Fig. 14). An in-place circuit can be created because both of the strong necessary conditions mentioned in the main text are met.

If Toffoli-count increase is allowed, 21 work qubits are required. In the further reduction step, Toffoli-depth value can be further reduced by 2. As a result, an in-place circuit with Toffoli-depth 6 and Toffoli-count 42 was created (Fig. 15).

B.2 Funtion f_2

The circuit created by optimizing the f_2 function based on Toffoli-count is the f_2-Z1 circuit (Fig. 16). At first glance, it looks similar to the χ-Z1 circuit, but if an increase in Toffoli-count is not allowed, the idea presented in the main text cannot reduce Toffoli-depth because it does not satisfy one of the necessary conditions. However, because an event occurs in which the state value that previously existed within the quantum circuit is restored in the middle of the circuit, Toffoli-depth can be reduced without increasing Toffoli-count: The fifth Toffoli gate in the f_2-Z1 circuit creates state K_2,

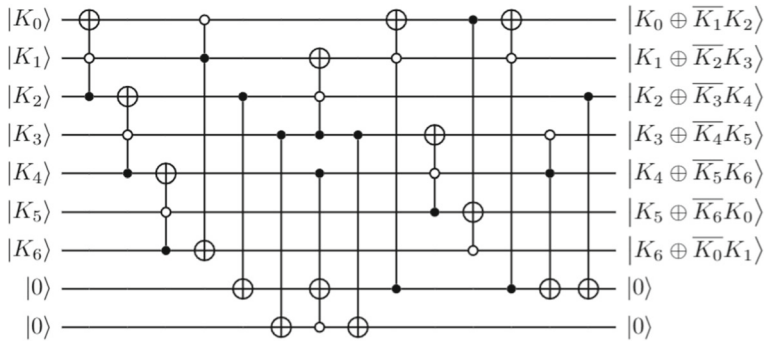

Fig. 14 f_1-Z2 circuit. This figure is presented to show an example of the proposed method being applied. Toffoli-depth value is still 9

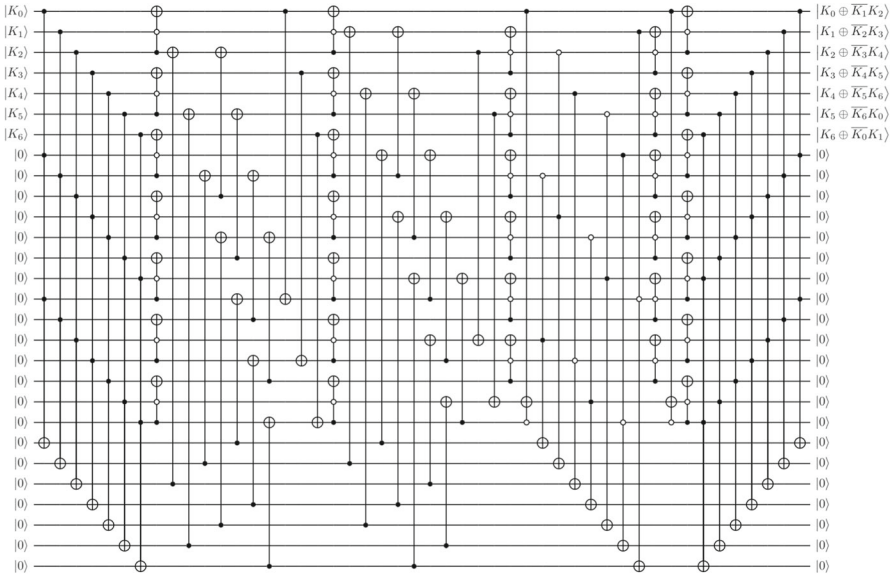

Fig. 15 f_1-Z3 circuit. 21 work qubits were used. Toffoli-depth value is 6 and Toffoli-count value is 42

and there is a gate (4th gate) that can use this state value. Therefore, a circuit in which two gates are obviously processed in parallel can be created (Fig. 17).

The f_2-Z3 circuit is an output circuit created when Toffoli-count value is allowed to increase (Fig. 18). It is quite similar in form to the χ-Z3 circuit. Toffoli-depth value is 4 and Toffoli-count value is 20.

B.3 Funtion f_3

The input circuit f_3-Z1 can have both Toffoli-depth and Toffoli-count reduced in the case of not increasing Toffoli-count (Fig. 19). As in the previous example, an event where the existing state value is restored occurs in the middle of the circuit so that

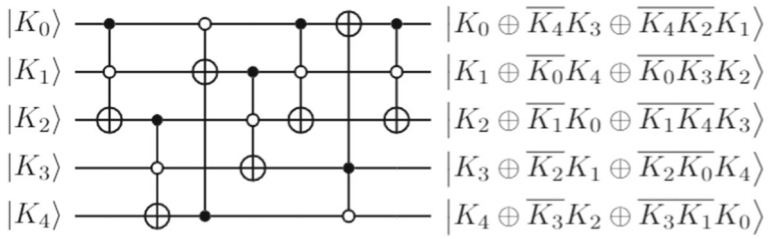

Fig. 16 f_2-Z1 circuit. Toffoli-count and Toffoli-depth are all 7

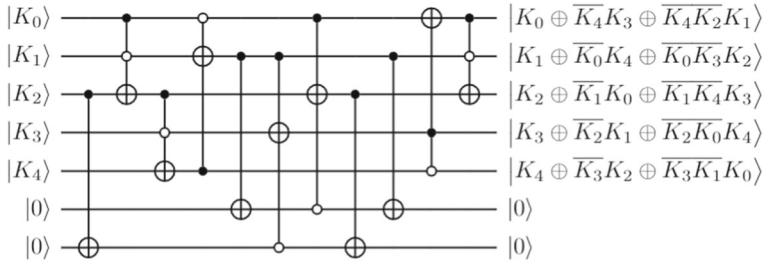

Fig. 17 f_2-Z2 circuit. Two work qubits are used, and Toffoli-depth is reduced by 1

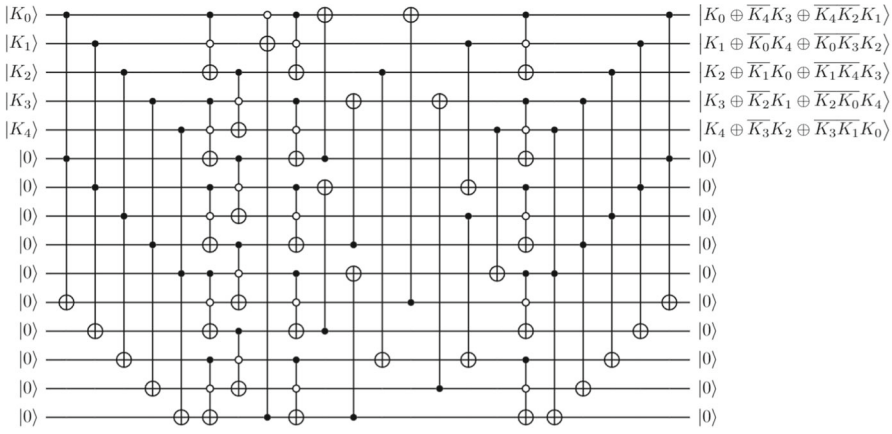

Fig. 18 f_2-Z3 circuit. Ten work qubits are used, and Toffoli-depth becomes 4 from 7

Toffoli-depth can be easily reduced (Fig. 20). (Global) Toffoli-count reduction is now possible as the location of the control line of one Toffoli gate is changed. As a result, Toffoli-count can also be reduced by 1 (Fig. 21). If Toffoli-count increase is allowed, a circuit with Toffoli-depth 5 and Toffoli-count 25 is created (Fig. 22).

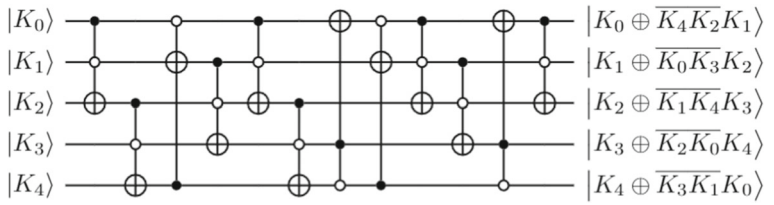

Fig. 19 f_3-Z1 circuit. Toffoli-count and Toffoli-depth are all 12

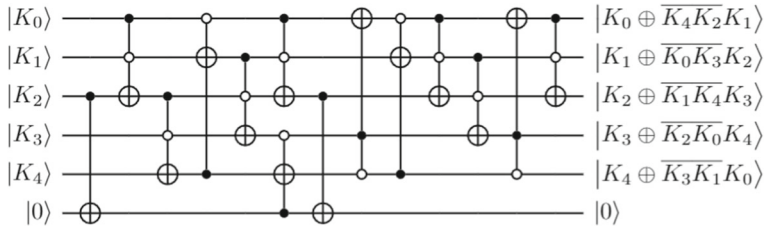

Fig. 20 f_3-Z2 circuit. Toffoli-depth is reduced by 1

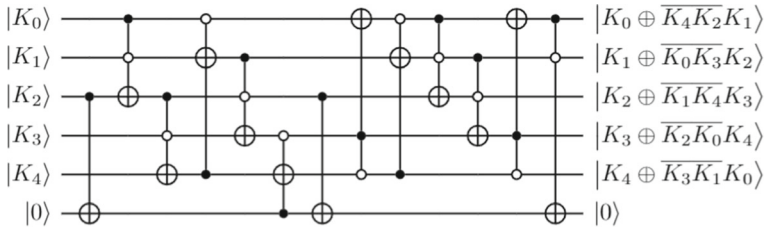

Fig. 21 f_3-Z3 circuit. Toffoli-count is reduced by 1 through the existing Toffoli-count reduction methods [22, 31–34]

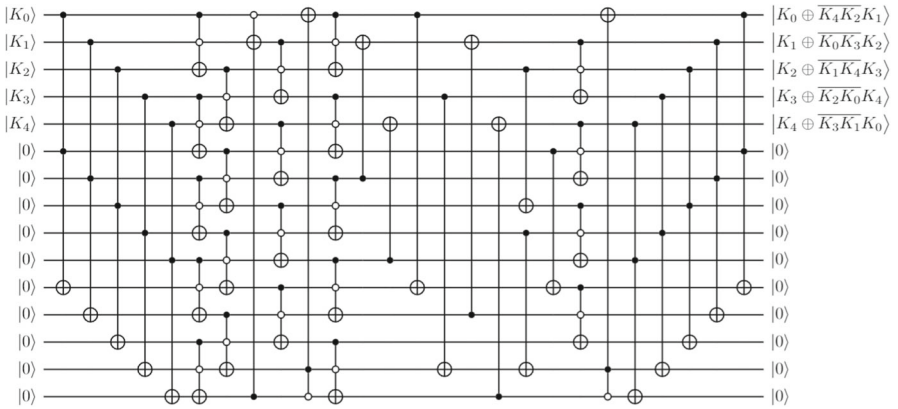

Fig. 22 f_3-Z4 circuit. Toffoli-depth becomes 5 from 12

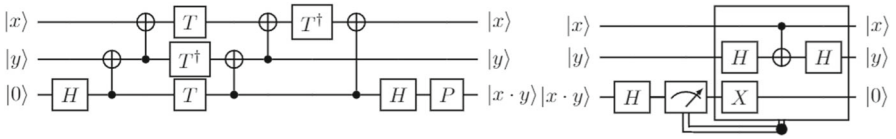

Fig. 23 Quantum AND and AND† gates [20]. Quantum AND gate consists of a C^2(-iX) gate and a P gate. The input state value of the target part should be 0. AND† gate uses an (intermediate) measurement meter and initializes the input value of the target part to 0

Appendix C: Conversion to MBQC-based quantum circuit

This section describes the result of the conversion of our χ circuits to MBQC (measurement-based quantum computation)-based circuits. If a quantum circuit is tried to design based on MBQC, (intermediate) measurement meters as well as quantum gates could be introduced [47]. Installing measurement meters in the middle of the circuit may lower DC or execution time, so it is worth trying to consider MBQC when constructing a circuit. Measurement meters are used in AND† gates [20]. AND gates are implemented as gates with T-depth 2 unless aided by any CWQs (Fig. 23). The product of the two binary input values x and y is returned in the target part of the gate. When AND and AND† gates are used instead of Toffoli gates, some necessary conditions are required: The input value of the target part of AND gate should be '0.' Symmetrically, AND† gate takes (x, y, x.y) and outputs (x,y,0).

We tried applying MBQC to χ-Z2 and χ-Z3 circuits, and two ideas were used. The first is to design the output pair of AND gate and the input pair of AND† gate to be different. For the output pair of AND gate is $(x, y, x \centerdot y)$, if the input pair of AND† gate is $(x', y, x \centerdot y)$, the target part $x \centerdot y$ of AND† gate could be not initialized to '0' in general. However, if $x \centerdot y = x' \centerdot y$, then the target part could be initialized. To use this idea, for a pair of inputs $(x', y, x \centerdot y)$ of AND† gate, x', y is set by two of output values, and $x \centerdot y$ is set by the product of those. Also, for the corresponding AND gate output pair $(x, y, x \centerdot y)$, x is set by the input or intermediate value, not the output value. For example, one of intermediate values $x \centerdot y = (\overline{K_0}K_1 \oplus \overline{K_0K_2}K_3)$ can be obtained through AND gate using $x = \overline{K_0}$ and $y = (K_1 \oplus \overline{K_2}K_3)$. This value is equivalent to $x' \centerdot y = (\overline{K_0} \oplus \overline{K_1}K_2) \centerdot (K_1 \oplus \overline{K_2}K_3)$.

The second idea is that some intermediate values are used to generate output values through Toffoli gates before they are initialized by AND† gates. Looking at the expression (C4), intermediate values can be used to create one of the output values $K_2 \oplus \overline{K_3}K_4$.

$$\begin{aligned}(K_4 \oplus \overline{K_0}K_1) \oplus (\overline{K_0}K_1 \oplus \overline{K_0K_2}K_3) = K_4 \oplus \overline{K_0K_2}K_3 \\ \Rightarrow K_2 \oplus (\overline{K_3} \oplus \overline{K_4}K_0)(K_4 \oplus \overline{K_0K_2}K_3) = K_2 \oplus \overline{K_3}K_4\end{aligned} \tag{C4}$$

These two ideas help to create χ circuits where MBQC is applied. As a result, χ-Z4 and χ-Z5 circuits are created (Figs. 24, 25). In the case of χ-Z4 circuit, two Toffoli gates have been changed to AND and AND† gates, respectively. This circuit has T-depth value 8, T-count value 35. One AND gate and one Toffoli gate share one Toffoli-depth.

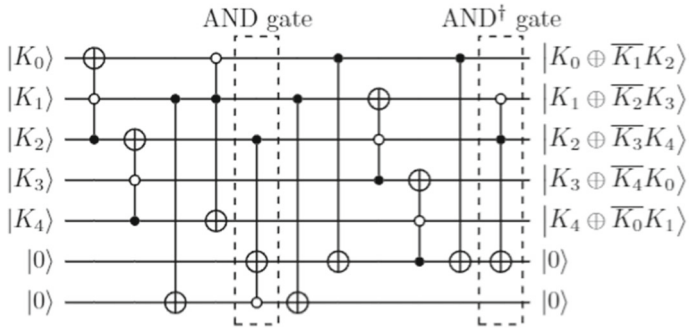

Fig. 24 An MBQC-based circuit where χ-Z2 circuit is converted (χ-Z4). Two Toffoli gates have been changed to AND and AND† gates, respectively

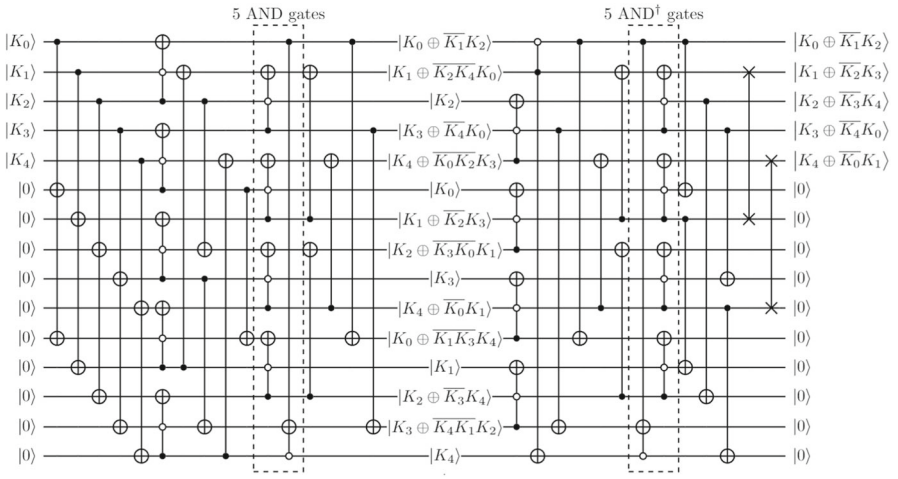

Fig. 25 A circuit where χ-Z3 circuit is converted by using MBQC (χ-Z5). 10 Toffoli gates and 5 AND and AND† gate pairs are used. The last two 2-qubit gates are SWAP gates

Half of 20 Toffoli gates in χ-Z3 circuit are replaced with AND or AND† gates in χ-Z5 circuit. As in χ-Z3, 10 CWQs were used, and Toffoli-depth is the same. However, this Toffoli-depth 4 includes AND-depth 2. The last two 2-qubit gates in χ-Z5 circuit are SWAP gates, which are used to properly align the states of the qubits.

For these two circuits, The positions and roles of some CNOT gates and Toffoli gates are swapped according to the two ideas mentioned above when comparing existing circuits. AND gates swap positions with CNOT gates to use clean qubits as the target qubits. AND† gates participate in the initialization of work qubits along with CNOT gates. The conversion method from general circuits to MBQC-based circuits requires additional research on what constraints are needed more.

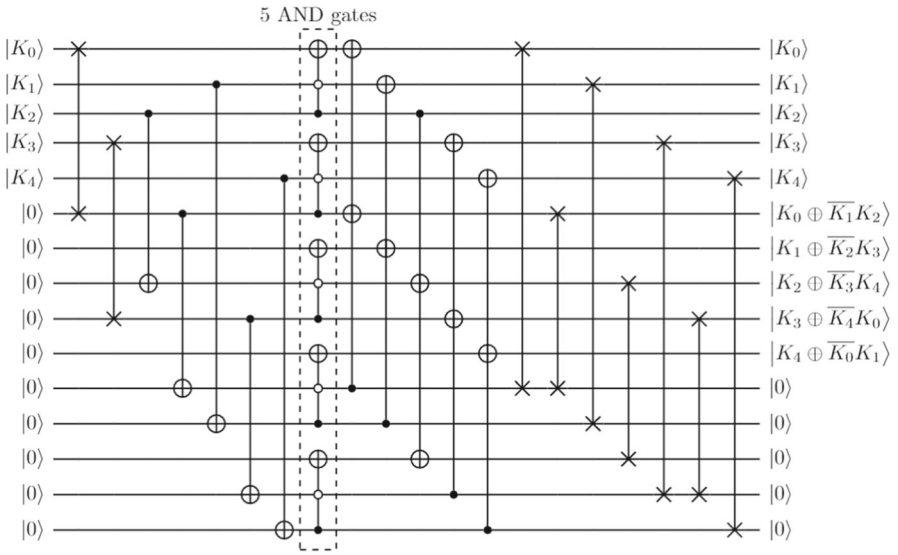

Fig. 26 An out-of-place version circuit for χ internal function [3, 6]. This circuit has not been specifically addressed in previous studies. Work qubits for AND gates are not displayed. Each AND gate's T-depth is 1

Appendix D: Out-of-place version circuit for χ function

In previous studies [3, 6], out-of-place version circuits for χ internal function were created. They attempted to reduce T-depth as much as possible while ignoring Width increment. Figure 26 shows an out-of-place version circuit for the χ internal function we make. Qubits with input values become garbage qubits (or DBQs) because their state values are no longer used in the next round in SHA3-256. Although it is not visible in the figure, five more CWQs exist, and they are recycled in each round. If one work qubit is added to each Quantum AND gate, it can be implemented as a circuit with T-depth value 1. In other words, it can become a 4-qubit gate with T-depth 1 [20].

As mentioned in the main text, 320 χ internal functions are processed simultaneously in parallel in one round. This simultaneous work is repeated 24 times for one message block. Therefore, assuming SHA3-256 processes one message block when out-of-place χ circuits are used, 38,400 DBQs are produced in total, and 3200 CWQs are recycled every round. That means creating a SHA3-256 circuit with T-depth 24 and Width 43,200 is possible (SHA3-256-Z0). This circuit is mentioned in Table 2.

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

Funding This work was partly supported by Institute of Information and communications Technology Planning and 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) and also partially supported 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).

Data Availability Data related to this paper are available on request from one of the authors, Jongheon Lee. No datasets were generated or analysed during the current study.

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

Declarations

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

References

1. Shahidi, S.M., Etemadi Borujeni, S.: A new method for reversible circuit synthesis using a simulated annealing algorithm and don't-cares. J. Comput. Electron. **20**(1), 718–734 (2021)
2. Fazel, K., Thornton, M.A., Rice, J.E.: ESOP-based Toffoli gate cascade generation. In: 2007 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing, pp. 206– 209. IEEE (2007)
3. Häner, T., Soeken, M.: Lowering the t-depth of quantum circuits via logic network optimization. ACM Trans. Quantum Comput. **3**(2), 1–15 (2022)
4. Testa, E., Soeken, M., Amarù, L., De Micheli, G.: Reducing the multiplicative complexity in logic networks for cryptography and security applications. In: 2019 56th ACM/IEEE Design Automation Conference (DAC), pp. 1– 6. IEEE (2019)
5. Testa, E., Soeken, M., Riener, H., Amaru, L., De Micheli, G.: A logic synthesis toolbox for reducing the multiplicative complexity in logic networks. In: 2020 Design, Automation and Test in Europe Conference and Exhibition (DATE), pp. 568– 573. IEEE (2020)
6. Meuli, G., Soeken, M., De Micheli, G.: Xor-and-inverter graphs for quantum compilation. NPJ Quantum Inf. **8**(1), 1–11 (2022)
7. Meuli, G., Soeken, M., Roetteler, M., Bjorner, N., De Micheli, G.: Reversible pebbling game for quantum memory management. In: 2019 Design, Automation and Test in Europe Conference and Exhibition (DATE), pp. 288– 291. IEEE (2019)
8. 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)
9. Nam, Y., Ross, N.J., Su, Y., Childs, A.M., Maslov, D.: Automated optimization of large quantum circuits with continuous parameters. NPJ Quantum Inf. **4**(1), 1–12 (2018)
10. Lee, J., Lee, S., Lee, Y.-S., Choi, D.: T-depth reduction method for efficient SHA-256 quantum circuit construction. IET Inf. Secur. (2022)
11. Maslov, D., Dueck, G.W., Miller, D.M., Negrevergne, C.: Quantum circuit simplification and level compaction. IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst. **27**(3), 436–444 (2008)
12. Saeedi, M., Markov, I.L.: Synthesis and optimization of reversible circuits-a survey. ACM Comput. Surv. (CSUR) **45**(2), 1–34 (2013)
13. 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)
14. Fowler, A.G., Stephens, A.M., Groszkowski, P.: High-threshold universal quantum computation on the surface code. Phys. Rev. A **80**(5), 052312 (2009)

15. 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)
16. Draper, T.G., Kutin, S.A., Rains, E.M., Svore, K.M.: A logarithmic-depth quantum carry-lookahead adder. quant-ph/0406142 (2004)
17. 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)
18. Amy, M., Matteo, O.D., Gheorghiu, V., Mosca, M., Parent, A., Schanck, J.: Estimating the cost of generic quantum pre-image attacks on SHA-2 and SHA-3. In: International Conference on Selected Areas in Cryptography, pp. 317– 337. Springer (2016)
19. Lee, J.: A study on T-depth and Toffoli-depth reduction techniques for efficient quantum circuit designs and their applications to hash functions. http://www.dcollection.net/handler/ust/200000651126 (2023)
20. Jaques, S., Naehrig, M., Roetteler, M., Virdia, F.: Implementing Grover oracles for quantum key search on AES and LowMC. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques, pp. 280– 310. Springer (2020)
21. 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)
22. Miller, D.M., Maslov, D., Dueck, G.W.: A transformation based algorithm for reversible logic synthesis. In: Proceedings 2003. Design Automation Conference (IEEE Cat. No. 03ch37451), pp. 318– 323. IEEE (2003)
23. Soeken, M., Tague, L., Dueck, G.W., Drechsler, R.: Ancilla-free synthesis of large reversible functions using binary decision diagrams. J. Symb. Comput. **73**, 1–26 (2016)
24. Zhu, W., Li, Z., Zhang, G., Pan, S., Zhang, W.: A reversible logical circuit synthesis algorithm based on decomposition of cycle representations of permutations. Int. J. Theor. Phys. **57**(8), 2466–2474 (2018)
25. Sasanian, Z., Saeedi, M., Sedighi, M., Zamani, M.S.: A cycle-based synthesis algorithm for reversible logic. In: 2009 Asia and South Pacific Design Automation Conference, pp. 745– 750. IEEE (2009)
26. Saeedi, M., Zamani, M.S., Sedighi, M., Sasanian, Z.: Reversible circuit synthesis using a cycle-based approach. ACM J. Emerg. Technol. Comput. Syst. (JETC) **6**(4), 1–26 (2010)
27. Yang, G., Xie, F., Song, X., Hung, W.N., Perkowski, M.A.: A constructive algorithm for reversible logic synthesis. In: 2006 IEEE International Conference on Evolutionary Computation, pp. 2416–2421. IEEE (2006)
28. Yang, G., Song, X., Hung, W.N., Xie, F., Perkowski, M.A.: Group theory based synthesis of binary reversible circuits. In: International Conference on Theory and Applications of Models of Computation, pp. 365– 374. Springer (2006)
29. Osman, M., Younes, A., Fahmy, M.H.: Integration of irreversible gates in reversible circuits using NCT library. IOSR J. Comput. Eng **14**, 69–79 (2013)
30. Fraleigh, J.B.: A First Course in Abstract Algebra. Pearson Education India, Chennai (2003)
31. Rahman, M.Z., Rice, J.E.: Templates for positive and negative control Toffoli networks. In: International Conference on Reversible Computation, pp. 125– 136. Springer (2014)
32. Maslov, D., Dueck, G.W., Miller, D.M.: Simplification of Toffoli networks via templates. In: 16th Symposium on Integrated Circuits and Systems Design, 2003. SBCCI 2003. Proceedings., pp. 53– 58. IEEE (2003)
33. Iwama, K., Kambayashi, Y., Yamashita, S.: Transformation rules for designing CNOT-based quantum circuits. In: Proceedings of the 39th Annual Design Automation Conference, pp. 419– 424 (2002)
34. Maslov, D., Young, C., Miller, D.M., Dueck, G.W.: Quantum circuit simplification using templates. In: Design, Automation and Test in Europe, pp. 1208– 1213. IEEE (2005)
35. Maslov, D.: Reversible logic synthesis benchmarks page. 2011. http://webhome.cs.uvic.ca/dmaslov (2020)
36. Dworkin, M.J., et al.: SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions. US Department of Commerce, National Institute of Standards and Technology (NIST) (2015)
37. Patel, K.N., Markov, I.L., Hayes, J.P.: Optimal synthesis of linear reversible circuits. Quantum Inf. Comput. **8**(3), 282–294 (2008)
38. Jiang, J., Sun, X., Teng, S.-H., Wu, B., Wu, K., Zhang, J.: Optimal space-depth trade-off of CNOT circuits in quantum logic synthesis. In: Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 213–229. SIAM (2020)
39. Miller, D.M., Soeken, M., Drechsler, R.: Mapping NCV circuits to optimized Clifford + T circuits. In: International Conference on Reversible Computation, pp. 163–175. Springer (2014)

40. 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)
41. Lee, J., Kang, Y., Lee, Y.-S., Chung, B., Choi, D.: MPMCT gate decomposition method reducing T-depth quickly in proportion to the number of work qubits. Quantum Inf. Process. **22**(10), 381 (2023)
42. 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)
43. Amy, M., Mosca, M.: T-count optimization and reed-muller codes. IEEE Trans. Inf. Theory **65**(8), 4771–4784 (2019)
44. Datta, K., Rathi, G., Sengupta, I., Rahaman, H.: Synthesis of reversible circuits using heuristic search method. In: 2012 25th International Conference on VLSI Design, pp. 328–333. IEEE (2012)
45. 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)
46. Wille, R., Lye, A., Niemann, P.: Checking reversibility of Boolean functions. In: Reversible Computation: 8th International Conference, RC 2016, Bologna, Italy, July 7–8, 2016, Proceedings 8, pp. 322–337. Springer (2016)
47. Raussendorf, R., Browne, D.E., Briegel, H.J.: Measurement-based quantum computation on cluster states. Phys. Rev. A **68**(2), 022312 (2003)