IEICE TRANSACTIONS

on Information and Systems

DOI:10.1587/transinf.2023LOP0003

Publicized:2024/05/10

This advance publication article will be replaced by the finalized version after proofreading.

A PUBLICATION OF THE INFORMATION AND SYSTEMS SOCIETY

The Institute of Electronics, Information and Communication Engineers

Kikai-Shinko-Kaikan Bldg., 5-8, Shibakoen 3 chome, Minato-ku, TOKYO, 105-0011 JAPAN

PAPER Special Section on Multiple-Valued Logic and VLSI Computing

New Bounds for Quick Computation of the Lower Bound on the Gate Count of Toffoli-Based Reversible Logic Circuits*

Takashi HIRAYAMA[†], *Member*, **Rin SUZUKI**[†], *Nonmember*, **Katsuhisa YAMANAKA**[†], *and* **Yasuaki NISHITANI**[†], *Members*

SUMMARY We present a time-efficient lower bound κ on the number of gates in Toffoli-based reversible circuits that represent a given reversible logic function. For the characteristic vector $\boldsymbol{s}$ of a reversible logic function, $\kappa(\boldsymbol{s})$ closely approximates $\sigma\text{-}lb(\boldsymbol{s})$, which is known as a relatively efficient lower bound in respect of evaluation time and tightness. The primary contribution of this paper is that κ enables fast computation while maintaining a tightness of the lower bound, approximately equal to $\sigma\text{-}lb$. We prove that the discrepancy between $\kappa(\boldsymbol{s})$ and $\sigma\text{-}lb(\boldsymbol{s})$ is at most one only, by providing upper and lower bounds on $\sigma\text{-}lb$ in terms of κ. Subsequently, we show that κ can be calculated more efficiently than $\sigma\text{-}lb$. An algorithm for $\kappa(\boldsymbol{s})$ with a complexity of $O(n)$ is presented, where n is the dimension of $\boldsymbol{s}$. Experimental results comparing κ and $\sigma\text{-}lb$ are also given. The results demonstrate that the two lower bounds are equal for most reversible functions, and that the calculation of κ is significantly faster than $\sigma\text{-}lb$ by several orders of magnitude.

key words: *reversible logic circuits, Toffoli gates, lower bound, logic minimization*

1. Introduction

Quantum computing has gained the interest of many researchers due to its promising high-performance computing as well as its potential low-energy consumption. The synthesis of reversible logic circuits is a fundamental part of the quantum logic field [2]. Figure 1 shows an example of a 4-bit reversible circuit, consisting of five gates $G_1, G_2, \ldots, G_5$. NOT, CNOT, and Toffoli gates are typically used to synthesize reversible logic circuits [3]–[6]. Fredkin and SWAP gates are also known [7]–[9] as other types of reversible logic gates. Figure 2 shows an example of Toffoli gates. The standard Toffoli with three bits, as depicted in Fig. 2(c), is often generalized to k-bit Toffoli such as Fig. 2(d). In this sense, NOT and CNOT can be regarded as 1-bit and 2-bit Toffoli gates, respectively. A k-bit Toffoli gate has $(k-1)$ control lines $x_1, x_2, \ldots, x_{k-1}$, denoted by $\bullet$, and a single target line x_k, denoted by $\oplus$. The target line maps x_k to $x_k \oplus x_1 x_2 \cdots x_{k-1}$ while the remaining lines propagate their signals without change. That is, a Toffoli gate inverts the value of the target line if all the control lines are assigned to 1. The set of these 1-, 2-, ..., k-bit Toffoli gates is referred to as the general Toffoli library. We discuss the reversible logic synthesis with the general Toffoli library.

The gate count (GC) is one of the most widely-used cost metrics for reversible logic circuits. There are many other technology-specific cost metrics [10]–[12], including the number of elementary gates, quantum cost, delay, depth, etc. However, their relevance will change depending on future developments in quantum technologies. For this reason, GC is commonly used as a fundamental cost metric. Although different gates require different resources in a more precise sense, this metric approximately reflects the complexity of circuits.

The upper and lower bounds on the GC have both theoretical and practical significance as a measure of the complexity of reversible circuits. Many studies have investigated upper [13], [14] and lower bounds for various classes of functions. Maslov et al. [15], Shende et al. [16], Soeken et al. [17], and Zakablukov [18] presented bounds on the GC of reversible circuits for the class of all n-input reversible functions under different constraints of ancillae, garbage outputs, gate types, etc. Popescu et al. [19] and Maslov [20] analyzed the bounds on the number of particular gate types in the NOT-CNOT-Toffoli library. Saeedi et al. [21] discussed the upper bounds for the classes of functions with various lengths of cycles.

For individual reversible functions, a circuit simplified by a heuristic synthesizer [3]–[5], [7], [9], [22], [23] can be seen as an upper bound for the exact minimum circuit. In this context, a lot of methods to determine upper bounds for given functions have already been studied.

Compared to upper bounds, works for the lower bounds for individual reversible functions are fewer. We deal with fully-specified n-variable reversible functions, and their circuit realization with the general Toffoli library without adding ancillae or garbage lines. Regarding the lower bounds for given functions, some bounds have been proposed based on the Positive Polarity Reed-Muller expressions [24], [25]. This type of bounds has practical applications as well as theoretical significance. They can be used to evaluate the complexity of functions in heuristic synthesis algorithms [26] or to reduce the search space in branch-and-bound algorithms [27]. For these applications, not only is the tightness of the bound essential, but low-cost computation is also crucial because the bound calculation is repeated extensively throughout the execution of algorithms.

$\sigma\text{-}lb(\boldsymbol{s})$, presented in the literature[25], is a lower

[†]The authors are with the Department of Systems Innovation Engineering, Iwate University, 4-3-5 Ueda, Morioka, Iwate, 020-8551 Japan.

*A preliminary work of this paper was presented at ISMVL 2023 [1]. This work was supported by JSPS KAKENHI Grant Number JP23K11027.

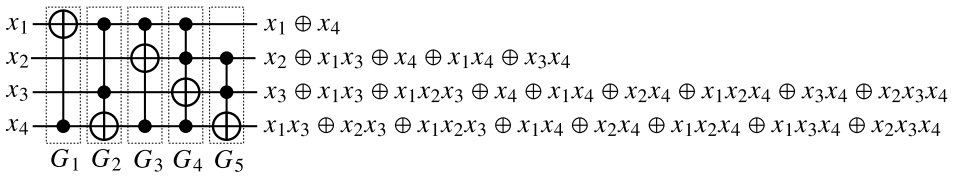


Fig. 1 Example of a reversible circuit

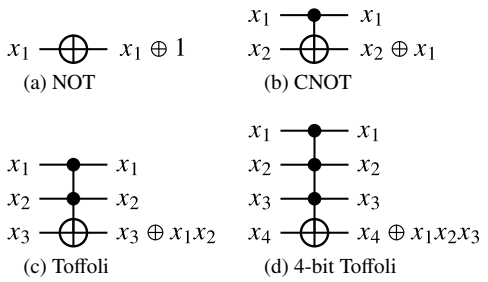


Fig. 2 Example of Toffoli gates.

bound on the GC of Toffoli-based reversible logic circuits of a given reversible logic function, where $\boldsymbol{s}$ is its characteristic vector. Although the bound σ-$lb(\boldsymbol{s})$ is tighter than the other ones proposed so far, its computation time is an obstacle to its application to synthesis algorithms of reversible circuits. Basically, the calculation of σ-$lb(\boldsymbol{s})$ involves an exhaustive search within an n-ary tree, where n denotes the dimension of the vector $\boldsymbol{s}$. For a characteristic vector $\boldsymbol{s}$ of a reversible function, the depth of the n-ary tree is proportional to n in maximum. It follows that the time complexity of the computation is roughly $O(n^{cn})$, where c is a constant. As such, the computation of σ-$lb(\boldsymbol{s})$ is time-consuming. In this paper, we investigate the upper and lower bounds on σ-$lb(\boldsymbol{s})$ and present a new lower bound $\tilde{\sigma}(\boldsymbol{s})$ and its faster version $\kappa(\boldsymbol{s})$. The value of $\kappa(\boldsymbol{s})$ is almost the same as σ-$lb(\boldsymbol{s})$; the discrepancy between $\kappa(\boldsymbol{s})$ and σ-$lb(\boldsymbol{s})$ is at most one only. Meanwhile, $\kappa(\boldsymbol{s})$ can be calculated quickly from $\boldsymbol{s}$ with a time complexity of $O(n)$.

The paper is organized as follows. Section 2 introduces the necessary preliminaries. It reviews the previous lower bound σ-lb and gives our new lower bound $\tilde{\sigma}$. To make a theoretical comparison, Sections 3 and 4 present bounds on σ-lb in terms of $\tilde{\sigma}$. In Section 5, a fast version of $\tilde{\sigma}$ is proposed and termed κ. The experimental results are reported in Section 6. We conclude in Section 7.

2. Preliminaries

To discuss lower bounds on the gate count (GC) of reversible circuits, we define a characteristic vector $\boldsymbol{\Lambda}(F)$ of a reversible function F by using positive polarity Reed-Muller expressions (PPRMs) and then briefly refer to the previous lower bound σ-lb [25]. Similar to "$\boldsymbol{\Lambda}$," we employ boldface to denote arithmetic functions that produce a vector.

It is known that any logic function can be uniquely represented by PPRM [28], [29]. For example, the logic function $x_1\bar{x}_2 + x_2$ is written as $x_1 \oplus x_2 \oplus x_1x_2$ in PPRM, where "$\oplus$" denotes the EXOR operation.

Definition 1: Let x^0 and x^1 denote 1 and x, respectively. The logical expression in the form:

$$\bigoplus_{0 \le i \le 2^n-1} a_i \cdot x_n^{i_n} x_{n-1}^{i_{n-1}} \cdots x_1^{i_1} \tag{1}$$

is a *positive polarity Reed-Muller expression (PPRM)*, where $(i_n, i_{n-1}, \ldots, i_1)$ is the binary representation of i, and $a_i \in \{0, 1\}$ is a constant. □

Definition 2: A multiple-output logic function $F : \{0,1\}^n \to \{0,1\}^n$ is *reversible* if and only if F is bijective. The number of product terms in the PPRM of a single-output logic function f is denoted by $\tau(f)$. Suppose that F is a reversible function of n input variables $(x_1, x_2, \ldots, x_n)$ and n output functions $(f_1, f_2, \ldots, f_n)$. $\boldsymbol{\Lambda}(F)$ is defined as the vector $[\tau(x_1 \oplus f_1), \tau(x_2 \oplus f_2), \ldots, \tau(x_n \oplus f_n)]$ and is called the *characteristic vector of* F. □

$\tau(x_i \oplus f_i)$ is called the Hamming distance in the Reed-Muller spectrum between x_i and f_i [26].

Example 1: Let F be the reversible function of the circuit in Fig. 1, in which the output functions are represented in PPRM form as $f_1 = x_1 \oplus x_4$, $f_2 = x_2 \oplus x_1x_3 \oplus x_4 \oplus x_1x_4 \oplus x_3x_4$, $f_3 = x_3 \oplus x_1x_3 \oplus x_1x_2x_3 \oplus x_4 \oplus x_1x_4 \oplus x_2x_4 \oplus x_1x_2x_4 \oplus x_3x_4 \oplus x_2x_3x_4$, $f_4 = x_1x_3 \oplus x_2x_3 \oplus x_1x_2x_3 \oplus x_1x_4 \oplus x_2x_4 \oplus x_1x_2x_4 \oplus x_1x_3x_4 \oplus x_2x_3x_4$. Then, $\tau(x_1 \oplus f_1) = \tau(\cancel{x_1} \oplus \cancel{x_1} \oplus x_4) = \tau(x_4) = 1$, $\tau(x_2 \oplus f_2) = \tau(\cancel{x_2} \oplus \cancel{x_2} \oplus x_1x_3 \oplus x_4 \oplus x_1x_4 \oplus x_3x_4 = \tau(x_1x_3 \oplus x_4 \oplus x_1x_4 \oplus x_3x_4) = 4$, $\tau(x_3 \oplus f_3) = \tau(\cancel{x_3} \oplus \cancel{x_3} \oplus x_1x_3 \oplus x_1x_2x_3 \oplus x_4 \oplus x_1x_4 \oplus x_2x_4 \oplus x_1x_2x_4 \oplus x_3x_4 \oplus x_2x_3x_4) = 8$, $\tau(x_4 \oplus f_4) = \tau(x_4 \oplus x_1x_3 \oplus x_2x_3 \oplus x_1x_2x_3 \oplus x_1x_4 \oplus x_2x_4 \oplus x_1x_2x_4 \oplus x_1x_3x_4 \oplus x_2x_3x_4) = 9$. Thus, the characteristic vector of F is $\boldsymbol{\Lambda}(F) = [1, 4, 8, 9]$. □

We present the following proposition on the maximum value of elements in $\boldsymbol{\Lambda}(F)$ because the number of product terms in the PPRM of an n-variable function is at most $2^n - 1$. In this paper, the term "proposition" indicates a theorem that is well known, elementary, or immediately obvious without proof.

Proposition 1: Let F be a reversible function of n variables. Every element in $\boldsymbol{\Lambda}(F)$ is a non-negative integer less than or equal to $2^n - 1$. □

Definition 3: Among all reversible circuits that realize a reversible function F, those with the exact minimum GC are called the *minimum circuits of* F. The GC of a minimum circuit of F is denoted by $\gamma(F)$. □

Theorem 1 (Lower Bound Theorem [25]): For any reversible function F,

$$\gamma(F) \geq \sigma\text{-}lb(\boldsymbol{\Lambda}(F))$$

holds. □

Theorem 1 shows that $\sigma\text{-}lb(\boldsymbol{\Lambda}(F))$ is a lower bound on the GC of a minimum circuit of F. The definition of $\sigma\text{-}lb$ is described below.

Definition 4: The set of non-negative integers is denoted by $\mathbb{N}_0$, and the n-dimensional space of $\mathbb{N}_0$ is denoted by $\mathbb{N}_0^n$. An *index* is an integer in $\{1, 2, \ldots, n\}$. Suppose that $\boldsymbol{s}$ is a vector $[s_1, s_2, \ldots, s_n] \in \mathbb{N}_0^n$, and a is an index. The function $\boldsymbol{\Phi}(\boldsymbol{s}, a) = [s'_1, s'_2, \ldots, s'_n]$ is defined as follows.

$$s'_i = \begin{cases} \lfloor s_i/2 \rfloor & (i = a) \\ \lceil s_i/2 \rceil & (\text{otherwise}) \end{cases}$$

We extend $\boldsymbol{\Phi}$ to allow it to accept a sequence of indices as its second argument:

$$\begin{cases} \boldsymbol{\Phi}(\boldsymbol{s}, \varepsilon) = \boldsymbol{s} \\ \boldsymbol{\Phi}(\boldsymbol{s}, a\alpha) = \boldsymbol{\Phi}(\boldsymbol{\Phi}(\boldsymbol{s}, a), \alpha), \end{cases}$$

where ε denotes the empty sequence and $\alpha \in \{1, 2, \ldots, n\}^*$ is a sequence of indices. □

Throughout this paper, the dimension of a vector is n.

Example 2: Let $\boldsymbol{s} = [1, 4, 8, 9]$. $\boldsymbol{\Phi}(\boldsymbol{s}, 1) = [\lfloor 1/2 \rfloor, \lceil 4/2 \rceil, \lceil 8/2 \rceil, \lceil 9/2 \rceil] = [0, 2, 4, 5]$, $\boldsymbol{\Phi}(\boldsymbol{s}, 2) = [\lceil 1/2 \rceil, \lfloor 4/2 \rfloor, \lceil 8/2 \rceil, \lceil 9/2 \rceil] = [1, 2, 4, 5]$, $\boldsymbol{\Phi}(\boldsymbol{s}, 3) = [\lceil 1/2 \rceil, \lceil 4/2 \rceil, \lfloor 8/2 \rfloor, \lceil 9/2 \rceil] = [1, 2, 4, 5]$, $\boldsymbol{\Phi}(\boldsymbol{s}, 4) = [\lceil 1/2 \rceil, \lceil 4/2 \rceil, \lceil 8/2 \rceil, \lfloor 9/2 \rfloor] = [1, 2, 4, 4]$. Let $\boldsymbol{s} = [1, 4, 8, 9]$ and the sequence of indices $\alpha = 1; 4; 2; 3; 4$. $\boldsymbol{\Phi}(\boldsymbol{s}, 1; 4; 2; 3; 4) = \boldsymbol{\Phi}(\boldsymbol{\Phi}([1, 4, 8, 9], 1), 4; 2; 3; 4) = \boldsymbol{\Phi}(\boldsymbol{\Phi}([0, 2, 4, 5], 4), 2; 3; 4) = \boldsymbol{\Phi}(\boldsymbol{\Phi}([0, 1, 2, 2], 2), 3; 4) = \boldsymbol{\Phi}(\boldsymbol{\Phi}([0, 0, 1, 1], 3), 4) = \boldsymbol{\Phi}([0, 0, 0, 1], 4) = [0, 0, 0, 0]$. □

Definition 5: For a vector $\boldsymbol{s} \in \mathbb{N}_0^n$, $\sigma\text{-}lb(\boldsymbol{s})$ is defined as

$$\sigma\text{-}lb(\boldsymbol{s}) = \min\{|\alpha| \mid \alpha \in \{1, 2, \ldots, n\}^*, \ \boldsymbol{\Phi}(\boldsymbol{s}, \alpha) = \boldsymbol{0}\},$$

where $|\alpha|$ denotes the length of the index sequence α, and $\boldsymbol{0}$ denotes the zero vector $[0, 0, \ldots, 0]$. □

$\sigma\text{-}lb(\boldsymbol{s})$ represents the minimum number of application times of $\boldsymbol{\Phi}$ to transform a given vector $\boldsymbol{s}$ into $\boldsymbol{0}$. Since the result of $\boldsymbol{\Phi}$ varies according to the index a provided as its second argument, many patterns of index sequences are searched to obtain $\sigma\text{-}lb(\boldsymbol{s})$. This process is computationally intensive, leading to a time complexity of $O(n^{cn})$, where c is a constant. In this paper, we introduce a simpler version of $\boldsymbol{\Phi}$, denoted as $\tilde{\boldsymbol{\Phi}}$, in order to estimate $\sigma\text{-}lb(\boldsymbol{s})$ with quick computation.

Definition 6: Suppose that $\boldsymbol{s}$ is a vector $[s_1, s_2, \ldots, s_n] \in \mathbb{N}_0^n$. $\iota(\boldsymbol{s})$ denotes the minimum index i in $\{1, 2, \ldots, n\}$ such that $s_i = 1$:

$$\iota(\boldsymbol{s}) = \begin{cases} 0 & (s_i \neq 1 \text{ for all } i \in \{1, 2, \ldots, n\}) \\ \min\{i \mid s_i = 1, \ i \in \{1, 2, \ldots, n\}\} & (\text{otherwise}) \end{cases}$$

Note that $\iota(\boldsymbol{s}) = 0$ if there are no 1's in the elements of $\boldsymbol{s}$.

The function $\tilde{\boldsymbol{\Phi}}(\boldsymbol{s}) = [s'_1, s'_2, \ldots, s'_n]$ is defined as follows.

$$s'_i = \begin{cases} 0 & (s_i = 1 \text{ and } i = \iota(\boldsymbol{s})) \\ 1 & (s_i = 1 \text{ and } i \neq \iota(\boldsymbol{s})) \\ \lfloor s_i/2 \rfloor & (\text{otherwise}) \end{cases}$$

□

For a given vector $\boldsymbol{s}$, the output of $\boldsymbol{\Phi}(\boldsymbol{s}, a)$ varies based on the chosen index a. In contrast, $\tilde{\boldsymbol{\Phi}}(\boldsymbol{s})$ consistently produces a unique vector. Hence, $\tilde{\boldsymbol{\Phi}}(\boldsymbol{s})$ is simpler in its functionality. Utilizing $\tilde{\boldsymbol{\Phi}}$, we define a function analogous to $\sigma\text{-}lb$.

Definition 7: For a vector $\boldsymbol{s} \in \mathbb{N}_0^n$, $\tilde{\sigma}(\boldsymbol{s})$ is defined as follows.

$$\tilde{\sigma}(\boldsymbol{s}) = \begin{cases} 0 & (\boldsymbol{s} = \boldsymbol{0}) \\ \tilde{\sigma}(\tilde{\boldsymbol{\Phi}}(\boldsymbol{s})) + 1 & (\text{otherwise}) \end{cases}$$

□

In other words, $\tilde{\sigma}(\boldsymbol{s})$ is defined by the number of applications of $\tilde{\boldsymbol{\Phi}}$ required until the nested application of $\tilde{\boldsymbol{\Phi}}(\tilde{\boldsymbol{\Phi}}(\cdots\tilde{\boldsymbol{\Phi}}(\boldsymbol{s})\cdots))$ reaches $\boldsymbol{0}$ for the first time.

Example 3: Let $\boldsymbol{s} = [1, 4, 8, 9]$. $\tilde{\sigma}(\boldsymbol{s}) = \tilde{\sigma}(\tilde{\boldsymbol{\Phi}}([1, 4, 8, 9])) + 1 = \tilde{\sigma}(\tilde{\boldsymbol{\Phi}}([0, 2, 4, 4])) + 2 = \tilde{\sigma}(\tilde{\boldsymbol{\Phi}}([0, 1, 2, 2])) + 3 = \tilde{\sigma}(\tilde{\boldsymbol{\Phi}}([0, 0, 1, 1])) + 4 = \tilde{\sigma}(\tilde{\boldsymbol{\Phi}}([0, 0, 0, 1])) + 5 = \tilde{\sigma}([0, 0, 0, 0]) + 5 = 5$. □

In the next two sections, we show that $\sigma\text{-}lb$ can be approximated by using $\tilde{\sigma}$. Specifically, we give the upper and lower bounds on $\sigma\text{-}lb$ in terms of $\tilde{\sigma}$. With these bounds, we prove that the value of $\tilde{\sigma}$ is theoretically almost equivalent to $\sigma\text{-}lb$.

3. Lower Bound on $\sigma\text{-}lb$

In this section, we compare $\tilde{\sigma}$ and $\sigma\text{-}lb$, and then prove that $\tilde{\sigma}$ is a lower bound on $\sigma\text{-}lb$. This result confirms that $\tilde{\sigma}$ is also a lower bound on the GC of reversible circuits.

Definition 8: For two vectors $\boldsymbol{s} = [s_1, s_2, \ldots, s_n]$ and $\boldsymbol{s}' = [s'_1, s'_2, \ldots, s'_n]$, we say $\boldsymbol{s} \leq \boldsymbol{s}'$ if $s_i \leq s'_i$ for all $i \in \{1, 2, \ldots, n\}$. Likewise, we say $\boldsymbol{s} \geq \boldsymbol{s}'$ if $s_i \geq s'_i$ for all $i \in \{1, 2, \ldots, n\}$. □

By the definitions of $\tilde{\sigma}$ and $\sigma\text{-}lb$, we have the following proposition.

Proposition 2: Suppose that $\boldsymbol{s}$ and $\boldsymbol{s}'$ are vectors in $\mathbb{N}_0^n$, and "$\asymp$" is one of the following relations: "=," "$\leq$," and "$\geq$". If $\boldsymbol{s} \asymp \boldsymbol{s}'$, then $\tilde{\sigma}(\boldsymbol{s}) \asymp \tilde{\sigma}(\boldsymbol{s}')$ and $\sigma\text{-}lb(\boldsymbol{s}) \asymp \sigma\text{-}lb(\boldsymbol{s}')$. □

Definition 9: Let $\boldsymbol{s} = [s_1, s_2, \ldots, s_n]$ and $\boldsymbol{s}' = [s'_1, s'_2, \ldots, s'_n]$ be vectors in $\mathbb{N}_0^n$. Let π be a permutation of the set $\{1, 2, \ldots, n\}$. If there exists a permutation π such that $s_{\pi(i)} = s'_i$ for all $i \in \{1, 2, \ldots, n\}$, then $\boldsymbol{s}$ *is P-equivalent to* $\boldsymbol{s}'$, which is denoted by $\boldsymbol{s} \sim \boldsymbol{s}'$. □

Proposition 3: Let $\boldsymbol{s}$ and $\boldsymbol{s}'$ be vectors in $\mathbb{N}_0^n$. If $\boldsymbol{s} \sim \boldsymbol{s}'$, then $\tilde{\sigma}(\boldsymbol{s}) = \tilde{\sigma}(\boldsymbol{s}')$ and $\sigma\text{-}lb(\boldsymbol{s}) = \sigma\text{-}lb(\boldsymbol{s}')$. □

Lemma 1: For a vector $\boldsymbol{s} \in \mathbb{N}_0^n$ and an index a, $\tilde{\sigma}(\boldsymbol{\Phi}(\boldsymbol{s}, a)) \geq \tilde{\sigma}(\tilde{\boldsymbol{\Phi}}(\boldsymbol{s}))$. □

Proof. Suppose that $\boldsymbol{s} = [s_1, s_2, \ldots, s_n]$, $\boldsymbol{s}' = \boldsymbol{\Phi}(\boldsymbol{s}, a) = [s'_1, s'_2, \ldots, s'_n]$, and $\tilde{\boldsymbol{s}} = \tilde{\boldsymbol{\Phi}}(\boldsymbol{s}) = [\tilde{s}_1, \tilde{s}_2, \ldots, \tilde{s}_n]$.

Let us consider the case of $\iota(\boldsymbol{s}) = 0$. There are no elements s_i such that $s_i = 1$. Then, $s'_i \geq \tilde{s}_i$ for all $i \in \{1, 2, \ldots, n\}$ since $\lceil s_i/2 \rceil \geq \lfloor s_i/2 \rfloor$. Hence, $\boldsymbol{s}' \geq \tilde{\boldsymbol{s}}$, and therefore $\tilde{\sigma}(\boldsymbol{s}') \geq \tilde{\sigma}(\tilde{\boldsymbol{s}})$ by Proposition 2.

Below, we assume that $\iota(\boldsymbol{s}) \neq 0$. Let $b = \iota(\boldsymbol{s})$. b may be equal to a. For all $i \in \{1, 2, \ldots, n\} \setminus \{a, b\}$, $s'_i \geq \tilde{s}_i$ since $s'_i = \lceil s_i/2 \rceil = 1 = \tilde{s}_i$ if $s_i = 1$, and $s'_i = \lceil s_i/2 \rceil \geq \lfloor s_i/2 \rfloor = \tilde{s}_i$ otherwise. For $i = b$, $\tilde{s}_b = 0$ by the definition of $\tilde{\boldsymbol{\Phi}}$. Then, $s'_b \geq \tilde{s}_b$ since s'_b is non-negative: $s'_b \geq 0 = \tilde{s}_b$. From the above argument, we have proven that $s'_i \geq \tilde{s}_i$ for all $i \in \{1, 2, \ldots, n\} \setminus \{a\}$. The remaining work to conclude $\boldsymbol{s}' \geq \tilde{\boldsymbol{s}}$ is to compare s'_a and $\tilde{s}_a$.
(Case of $a = b$): Since $s'_b \geq \tilde{s}_b$, we have $s'_a \geq \tilde{s}_a$. Hence, $s'_i \geq \tilde{s}_i$ for all $i \in \{1, 2, \ldots, n\}$. Since $\boldsymbol{s}' \geq \tilde{\boldsymbol{s}}$, we have $\tilde{\sigma}(\boldsymbol{s}') \geq \tilde{\sigma}(\tilde{\boldsymbol{s}})$ by Proposition 2.
(Case of $s_a \neq 1$ and $a \neq b$): By the definitions of $\boldsymbol{\Phi}$ and $\tilde{\boldsymbol{\Phi}}$, $s'_a = \lfloor s_a/2 \rfloor = \tilde{s}_a$. Hence, $s'_i \geq \tilde{s}_i$ for all $i \in \{1, 2, \ldots, n\}$. Since $\boldsymbol{s}' \geq \tilde{\boldsymbol{s}}$, we have $\tilde{\sigma}(\boldsymbol{s}') \geq \tilde{\sigma}(\tilde{\boldsymbol{s}})$ by Proposition 2.
(Case of $s_a = 1$ and $a \neq b$): By the definitions of $\boldsymbol{\Phi}$ and $\tilde{\boldsymbol{\Phi}}$, $s'_a = \lfloor 1/2 \rfloor = 0$ and $\tilde{s}_a = 1$ while $s'_b = \lceil 1/2 \rceil = 1$ and $\tilde{s}_b = 0$. Let $\boldsymbol{s}''$ be the vector obtained by swapping s'_a and s'_b in $\boldsymbol{s}'$. Then, $\boldsymbol{s}'' \geq \tilde{\boldsymbol{s}}$, and therefore $\tilde{\sigma}(\boldsymbol{s}'') \geq \tilde{\sigma}(\tilde{\boldsymbol{s}})$ by Proposition 2. Since $\boldsymbol{s}' \sim \boldsymbol{s}''$, $\tilde{\sigma}(\boldsymbol{s}') = \tilde{\sigma}(\boldsymbol{s}'')$ by Proposition 3. Thus, we have $\tilde{\sigma}(\boldsymbol{s}') = \tilde{\sigma}(\boldsymbol{s}'') \geq \tilde{\sigma}(\tilde{\boldsymbol{s}})$. □

The next proposition follows immediately from Definition 7. This will be used in the proofs of Lemma 2 and some lemmas in the later sections.

Proposition 4: For a vector $\boldsymbol{s} \in \mathbb{N}_0^n \setminus \{\boldsymbol{0}\}$, $\tilde{\sigma}(\tilde{\boldsymbol{\Phi}}(\boldsymbol{s})) = \tilde{\sigma}(\boldsymbol{s}) - 1$. □

Lemma 2: Let $\boldsymbol{s} \in \mathbb{N}_0^n$ and $\alpha \in \{1, 2, \ldots, n\}^*$. If $\boldsymbol{\Phi}(\boldsymbol{s}, \alpha) = \boldsymbol{0}$, then $|\alpha| \geq \tilde{\sigma}(\boldsymbol{s})$. □

Proof. The proof is by the mathematical induction on the length of α. If $\boldsymbol{\Phi}(\boldsymbol{s}, \varepsilon) = \boldsymbol{0}$, then $\boldsymbol{s} = \boldsymbol{0}$, and therefore $\tilde{\sigma}(\boldsymbol{s}) = 0$. Thus, the lemma holds for $|\alpha| = 0$.

Assume that the lemma holds for $|\alpha'| = k$, that is, if $\boldsymbol{\Phi}(\boldsymbol{s}', \alpha') = \boldsymbol{0}$ holds for a vector $\boldsymbol{s}'$ in $\mathbb{N}_0^n$ and an index sequence α' such that $|\alpha'| = k$, then $|\alpha'| = k \geq \tilde{\sigma}(\boldsymbol{s}')$. Using this assumption, we prove the lemma for $|\alpha| = k + 1$. Suppose that $\boldsymbol{s}$ is a vector in $\mathbb{N}_0^n$ and α is an index sequence with a length of $k+1$ ($|\alpha| = k+1$). Then, α can be written in the concatenation of certain a and α' as $\alpha = a\alpha'$, where $a \in \{1, 2, \ldots, n\}$ and $|\alpha'| = k$. Let $\boldsymbol{s}'$ be $\boldsymbol{\Phi}(\boldsymbol{s}, a)$. If $\boldsymbol{\Phi}(\boldsymbol{s}, a\alpha') = \boldsymbol{0}$, then we have $\boldsymbol{\Phi}(\boldsymbol{s}, a\alpha') = \boldsymbol{\Phi}(\boldsymbol{\Phi}(\boldsymbol{s}, a), \alpha') = \boldsymbol{\Phi}(\boldsymbol{s}', \alpha') = \boldsymbol{0}$ by the definition of $\boldsymbol{\Phi}$. By using the induction hypothesis, $k \geq \tilde{\sigma}(\boldsymbol{s}')$ holds, and therefore $|\alpha| = k + 1 \geq \tilde{\sigma}(\boldsymbol{s}') + 1$. By Lemma 1 and Proposition 4, $\tilde{\sigma}(\boldsymbol{s}') \geq \tilde{\sigma}(\tilde{\boldsymbol{\Phi}}(\boldsymbol{s})) = \tilde{\sigma}(\boldsymbol{s}) - 1$. Then, we have $|\alpha| \geq \tilde{\sigma}(\boldsymbol{s}') + 1 \geq \tilde{\sigma}(\boldsymbol{s}) - 1 + 1 = \tilde{\sigma}(\boldsymbol{s})$. Thus, the inductive step was proved. □

Theorem 2: For a vector $\boldsymbol{s} \in \mathbb{N}_0^n$,

$$\sigma\text{-}lb(\boldsymbol{s}) \geq \tilde{\sigma}(\boldsymbol{s})$$

holds. □

Proof. By the definition of $\sigma\text{-}lb$, there exists a sequence α such that $\boldsymbol{\Phi}(\boldsymbol{s}, \alpha) = \boldsymbol{0}$ and $|\alpha| = \sigma\text{-}lb(\boldsymbol{s})$. Then, $|\alpha| \geq \tilde{\sigma}(\boldsymbol{s})$ holds by Lemma 2. Thus, we have $\sigma\text{-}lb(\boldsymbol{s}) = |\alpha| \geq \tilde{\sigma}(\boldsymbol{s})$. □

Theorem 2 shows that $\tilde{\sigma}$ is a lower bound on $\sigma\text{-}lb$. By Theorems 1 and 2, we can see that $\tilde{\sigma}$ is also a lower bound on the GC of a circuit of a reversible function. This fact is expressed as Theorem 3 below.

Theorem 3: For any reversible function F,

$$\gamma(F) \geq \tilde{\sigma}(\boldsymbol{\Lambda}(F))$$

holds. □

Example 4: As we have seen in Example 1, the characteristic vector of the reversible function F in Fig. 1 is $\boldsymbol{\Lambda}(F) = [1, 4, 8, 9]$. In Example 3, we obtained $\tilde{\sigma}([1, 4, 8, 9]) = 5$. Thus, $\gamma(F) \geq 5$ follows from Theorem 3, implying that five or more gates are required to realize F. Meanwhile, the circuit in Fig. 1 is realized by five gates. In this case, we can conclude that Fig. 1 is the exact minimum circuit of F and that $\gamma(F) = 5$. □

4. Upper Bound on $\sigma\text{-}lb$

In this section, we discuss the proximity of $\tilde{\sigma}$ to $\sigma\text{-}lb$, by providing an upper bound on $\sigma\text{-}lb$ in terms of $\tilde{\sigma}$.

Definition 10: We call a vector $\boldsymbol{s} \in \mathbb{N}_0^n$ a *power-of-two vector* if every element of $\boldsymbol{s}$ is 0 or a power of two: $2^0, 2^1, 2^2, \cdots$. A power-of-two vector except $\boldsymbol{0}$ (the zero vector) is called a *non-zero power-of-two vector*. □

Example 5: Let $\boldsymbol{s}_0 = [0, 0, 0, 0]$, $\boldsymbol{s}_1 = [2, 1, 4, 0] = [2^1, 2^0, 2^2, 0]$. $\boldsymbol{s}_0$ and $\boldsymbol{s}_1$ are power-of-two vectors. $\boldsymbol{s}_1$ is a non-zero power-of-two vector.

Lemma 3: If $\boldsymbol{r}$ is a power-of-two vector and a is an index, then $\boldsymbol{\Phi}(\boldsymbol{r}, a)$ is a power-of-two vector. □

Proof. Let $\boldsymbol{r}' = \boldsymbol{\Phi}(\boldsymbol{r}, a) = [r'_1, r'_2, \ldots, r'_n]$. $\lceil r_i/2 \rceil = \lfloor r_i/2 \rfloor = r_i/2$ if r_i is even. Since $\boldsymbol{r}$ is a power-of-two vector, a possible odd number in $\boldsymbol{r}$ is 1 only. Therefore, we have the following equation by the definition of $\boldsymbol{\Phi}$.

$$r'_i = \begin{cases} 0 & (r_i = 1 \text{ and } i = a) \\ 1 & (r_i = 1 \text{ and } i \neq a) \\ r_i/2 & (\text{otherwise}) \end{cases}$$

From this equation, we see that (i) if $r_i = 0$, then $r'_i = r_i/2 = 0$, (ii) if $r_i = 1$, then r'_i is 0 or 1 ($= 2^0$), and (iii) if $r_i \in \{2^1, 2^2, \ldots\}$, then $r'_i = r_i/2$ is a power of two. Since every r'_i is 0 or a power of two, $\boldsymbol{r}'$ is a power-of-two vector. □

Lemma 4: Let $\boldsymbol{r}$ be a power-of-two vector. There exists an index a such that $\boldsymbol{\Phi}(\boldsymbol{r}, a) = \tilde{\boldsymbol{\Phi}}(\boldsymbol{r})$. □

Proof. Suppose that $\boldsymbol{r} = [r_1, r_2, \ldots, r_n]$, $\boldsymbol{r}' = \boldsymbol{\Phi}(\boldsymbol{r}, a) = [r'_1, r'_2, \ldots, r'_n]$, and $\tilde{\boldsymbol{r}} = \tilde{\boldsymbol{\Phi}}(\boldsymbol{r}) = [\tilde{r}_1, \tilde{r}_2, \ldots, \tilde{r}_n]$. From the proof of Lemma 3, the following equation holds.

$$r'_i = \begin{cases} 0 & (r_i = 1 \text{ and } i = a) \\ 1 & (r_i = 1 \text{ and } i \neq a) \\ r_i/2 & (\text{otherwise}) \end{cases}$$

Similarly, the following equation holds by the definition of $\tilde{\boldsymbol{\Phi}}$.

$$\tilde{r}_i = \begin{cases} 0 & (r_i = 1 \text{ and } i = \iota(\boldsymbol{r})) \\ 1 & (r_i = 1 \text{ and } i \neq \iota(\boldsymbol{r})) \\ r_i/2 & (\text{otherwise}) \end{cases}$$

(Case of $\iota(\boldsymbol{r}) = 0$): In this case, there are no elements r_i such that $r_i = 1$. Then, $r'_i = r_i/2 = \tilde{r}_i$ for all $i \in \{1, 2, \ldots, n\}$. Thus, $\boldsymbol{r}' = \tilde{\boldsymbol{r}}$. The lemma holds for an arbitrary a.
(Case of $\iota(\boldsymbol{r}) \neq 0$): By choosing $\iota(\boldsymbol{r})$ as a, we have $r'_i = \tilde{r}_i$ for all $i \in \{1, 2, \ldots, n\}$. Thus, $\boldsymbol{r}' = \tilde{\boldsymbol{r}}$. □

Lemma 5: Let $\boldsymbol{r}$ be a non-zero power-of-two vector. There exists an index a such that $\tilde{\sigma}(\boldsymbol{\Phi}(\boldsymbol{r}, a)) = \tilde{\sigma}(\boldsymbol{r}) - 1$. □

Proof. By Lemma 4 and Proposition 4, there exists an index a such that $\tilde{\sigma}(\boldsymbol{\Phi}(\boldsymbol{r}, a)) = \tilde{\sigma}(\tilde{\boldsymbol{\Phi}}(\boldsymbol{r})) = \tilde{\sigma}(\boldsymbol{r}) - 1$. □

Lemma 6: For a power-of-two vector $\boldsymbol{r}$, $\sigma\text{-}lb(\boldsymbol{r}) = \tilde{\sigma}(\boldsymbol{r})$. □

Proof. Since $\sigma\text{-}lb(\boldsymbol{r}) \geq \tilde{\sigma}(\boldsymbol{r})$ holds by Theorem 2, we will prove that

$$\sigma\text{-}lb(\boldsymbol{r}) \leq \tilde{\sigma}(\boldsymbol{r}) \tag{2}$$

by the mathematical induction on $\tilde{\sigma}(\boldsymbol{r})$.

If $\tilde{\sigma}(\boldsymbol{r}) = 0$, then $\boldsymbol{r} = \boldsymbol{0}$, and therefore $\sigma\text{-}lb(\boldsymbol{r}) = 0$. Thus, Equation (2) holds for $\tilde{\sigma}(\boldsymbol{r}) = 0$.

Assume that Equation (2) holds for any power-of-two vector $\boldsymbol{r}'$ such that $\tilde{\sigma}(\boldsymbol{r}') = k$, i.e., $\sigma\text{-}lb(\boldsymbol{r}') \leq \tilde{\sigma}(\boldsymbol{r}') = k$. Then, by the definition of $\sigma\text{-}lb$, there exists an index sequence α' such that $|\alpha'| = k$ and $\boldsymbol{\Phi}(\boldsymbol{r}', \alpha') = \boldsymbol{0}$. Using this assumption, we prove Equation (2) for $\tilde{\sigma}(\boldsymbol{r}) = k + 1$. Suppose that $\boldsymbol{r} = [r_1, r_2, \ldots, r_n]$ is a non-zero power-of-two vector such that $\tilde{\sigma}(\boldsymbol{r}) = k + 1$. By Lemma 3, $\boldsymbol{\Phi}(\boldsymbol{r}, a)$ is a power-of-two vector. Then, let $\boldsymbol{r}' = \boldsymbol{\Phi}(\boldsymbol{r}, a)$, where a is the index in Lemma 5. By Lemma 5, $\tilde{\sigma}(\boldsymbol{r}') = \tilde{\sigma}(\boldsymbol{r}) - 1 = (k+1)-1 = k$. By using the induction hypothesis, there exists an index sequence α' such that $|\alpha'| = k$ and $\boldsymbol{\Phi}(\boldsymbol{r}', \alpha') = \boldsymbol{0}$. Then, $\boldsymbol{0} = \boldsymbol{\Phi}(\boldsymbol{r}', \alpha') = \boldsymbol{\Phi}(\boldsymbol{\Phi}(\boldsymbol{r}, a), \alpha') = \boldsymbol{\Phi}(\boldsymbol{r}, a\alpha')$ by the definition of $\boldsymbol{\Phi}$. Since $|\alpha'| = k$, we have $|a\alpha'| = k + 1$. By the definition of $\sigma\text{-}lb$, $\sigma\text{-}lb(\boldsymbol{r}) \leq |a\alpha'| = k + 1 = \tilde{\sigma}(\boldsymbol{r})$. Thus, the inductive step was proved. □

Definition 11: Suppose that $\boldsymbol{s}$ is a vector $[s_1, s_2, \ldots, s_n] \in \mathbb{N}_0^n$. The function $\boldsymbol{R}(\boldsymbol{s}) = [r_1, r_2, \ldots, r_n]$ is defined as follows, where $L(s_i)$ denotes the bit length of the binary representation of s_i, i.e., $L(s_i) = \lceil \log_2(s_i + 1) \rceil$.

$$r_i = \begin{cases} 0 & (s_i = 0) \\ s_i & (s_i \text{ is a power of 2: } 2^0, 2^1, 2^2, \ldots) \\ 2^{L(s_i)} & (\text{otherwise}) \end{cases}$$

□

Example 6: Let $\boldsymbol{s} = [1, 0, 2, 5]$. $\boldsymbol{R}(\boldsymbol{s}) = [1, 0, 2, 8]$.

$\boldsymbol{R}(\boldsymbol{s})$ rounds up all $s_i \neq 0$ to the power of two. Therefore, we have the following proposition.

Proposition 5: For a vector $\boldsymbol{s} \in \mathbb{N}_0^n$, $\boldsymbol{R}(\boldsymbol{s})$ is a power-of-two vector. □

Theorem 4: For a vector $\boldsymbol{s} \in \mathbb{N}_0^n$,

$$\sigma\text{-}lb(\boldsymbol{s}) \leq \tilde{\sigma}(\boldsymbol{R}(\boldsymbol{s}))$$

holds. □

Proof. Let $\boldsymbol{r} = \boldsymbol{R}(\boldsymbol{s})$. Since $\boldsymbol{s} \leq \boldsymbol{r}$ by the definition of $\boldsymbol{R}$, $\sigma\text{-}lb(\boldsymbol{s}) \leq \sigma\text{-}lb(\boldsymbol{r})$ by Proposition 2. By Proposition 5, $\boldsymbol{r}$ is a power-of-two vector. Then, $\sigma\text{-}lb(\boldsymbol{r}) = \tilde{\sigma}(\boldsymbol{r})$ by Lemma 6. Thus, we have $\sigma\text{-}lb(\boldsymbol{s}) \leq \sigma\text{-}lb(\boldsymbol{r}) = \tilde{\sigma}(\boldsymbol{r})$. □

Theorem 4 gives an upper bound on $\sigma\text{-}lb$ in terms of $\tilde{\sigma}$.

Using the bounds in Theorems 2 and 4, we evaluate the proximity of $\tilde{\sigma}$ to $\sigma\text{-}lb$. The result will be given as Theorem 5.

Lemma 7: For a vector $\boldsymbol{s} \in \mathbb{N}_0^n$, $\tilde{\boldsymbol{\Phi}}(\boldsymbol{R}(\boldsymbol{s})) \leq \boldsymbol{s}$. □

Proof. Suppose that $[s_1, s_2, \ldots, s_n] = \boldsymbol{s}$, $\boldsymbol{r} = \boldsymbol{R}(\boldsymbol{s}) = [r_1, r_2, \ldots, r_n]$, and $\tilde{\boldsymbol{r}} = \tilde{\boldsymbol{\Phi}}(\boldsymbol{r}) = [\tilde{r}_1, \tilde{r}_2, \ldots, \tilde{r}_n]$. By the definition of $\boldsymbol{R}$, the following equation holds.

$$r_i = \begin{cases} 0 & (s_i = 0) \\ 1 & (s_i = 1) \\ s_i & (s_i \geq 2 \text{ and } s_i \text{ is a power of 2}) \\ 2^{L(s_i)} & (\text{otherwise}) \end{cases}$$

Since $\boldsymbol{r}$ is a power-of-two vector by Proposition 5, we have the following equation from the above equation and the definition of $\tilde{\boldsymbol{\Phi}}$.

$$\tilde{r}_i = \begin{cases} 0 & (s_i = 0) \\ 0 & (s_i = 1 \text{ and } i = \iota(\boldsymbol{s})) \\ 1 & (s_i = 1 \text{ and } i \neq \iota(\boldsymbol{s})) \\ s_i/2 & (s_i \geq 2 \text{ and } s_i \text{ is a power of 2}) \\ 2^{L(s_i)-1} & (\text{otherwise}) \end{cases}$$

It can be observed that $\tilde{r}_i \leq s_i$ holds for all the cases above. Thus, $\tilde{\boldsymbol{r}} \leq \boldsymbol{s}$. □

Lemma 8: For a vector $\boldsymbol{s} \in \mathbb{N}_0^n$, $\tilde{\sigma}(\boldsymbol{R}(\boldsymbol{s})) \leq \tilde{\sigma}(\boldsymbol{s}) + 1$. □

Proof. The lemma clearly holds for $\boldsymbol{s} = \boldsymbol{0}$. In the following, we prove the lemma under the assumption of $\boldsymbol{s} \neq \boldsymbol{0}$. Let $\boldsymbol{r} = \boldsymbol{R}(\boldsymbol{s})$. By Proposition 4, $\tilde{\sigma}(\boldsymbol{r}) - 1 = \tilde{\sigma}(\tilde{\boldsymbol{\Phi}}(\boldsymbol{r}))$. By Lemma 7 and Proposition 2, $\tilde{\sigma}(\tilde{\boldsymbol{\Phi}}(\boldsymbol{r})) \leq \tilde{\sigma}(\boldsymbol{s})$. Thus, we have $\tilde{\sigma}(\boldsymbol{r}) - 1 = \tilde{\sigma}(\tilde{\boldsymbol{\Phi}}(\boldsymbol{r})) \leq \tilde{\sigma}(\boldsymbol{s})$. □

Theorem 5: For a vector $\boldsymbol{s} \in \mathbb{N}_0^n$,

$$\tilde{\sigma}(\boldsymbol{s}) \leq \sigma\text{-}lb(\boldsymbol{s}) \leq \tilde{\sigma}(\boldsymbol{s}) + 1$$

holds. □

Proof. $\tilde{\sigma}(\boldsymbol{s}) \leq \sigma\text{-}lb(\boldsymbol{s})$ by Theorem 2. Meanwhile, $\sigma\text{-}lb(\boldsymbol{s}) \leq \tilde{\sigma}(\boldsymbol{R}(\boldsymbol{s})) \leq \tilde{\sigma}(\boldsymbol{s})+1$ by Theorem 4 and Lemma 8. Thus, we have the theorem. □

Theorem 5 guarantees that the maximum difference between $\tilde{\sigma}(\boldsymbol{s})$ and $\sigma\text{-}lb(\boldsymbol{s})$ is only one. This indicates that $\tilde{\sigma}(\boldsymbol{s})$ closely approximates $\sigma\text{-}lb(\boldsymbol{s})$.

5. Quick Calculation of $\tilde{\sigma}$

In this section, a quick calculation of $\tilde{\sigma}(\boldsymbol{\Lambda}(F))$ is discussed. For an n-variable reversible function F, the dimension of $\boldsymbol{\Lambda}(F)$ is n by Definition 2, and the elements in $\boldsymbol{\Lambda}(F)$ are bounded by $2^n - 1$ by Proposition 1. Since $\tilde{\boldsymbol{\Phi}}$ halves the elements of a vector, $\tilde{\sigma}(\boldsymbol{\Lambda}(F))$ can be determined by applying $\tilde{\boldsymbol{\Phi}}$ at most cn times, where c is a certain constant. Moreover, the complexity of $\tilde{\boldsymbol{\Phi}}(\boldsymbol{s})$ is $O(n)$ by Definition 6. Hence, the time complexity of the straightforward computation of $\tilde{\sigma}(\boldsymbol{\Lambda}(F))$ results in $O(n^2)$. To improve this, we propose a quick calculation of $\tilde{\sigma}$ based on some arithmetic functions for the bit length. This optimized version achieves a time complexity of $O(n)$.

According to Definition 7, $\tilde{\sigma}(\boldsymbol{s})$ denotes the number of necessary applications of $\tilde{\boldsymbol{\Phi}}$ in the nested application of $\tilde{\boldsymbol{\Phi}}(\tilde{\boldsymbol{\Phi}}(\cdots\tilde{\boldsymbol{\Phi}}(\boldsymbol{s})\cdots))$ for the conversion from $\boldsymbol{s}$ to $\boldsymbol{0}$. Interpreting non-negative integers in binary representation, we can regarded the division $\lfloor s_i/2 \rfloor$ in $\tilde{\boldsymbol{\Phi}}(\boldsymbol{s})$ as a right-shift operation. Therefore, the value of $\tilde{\sigma}(\boldsymbol{s})$ depends on the bit length of the elements in $\boldsymbol{s}$. Utilizing this insight, we propose a method to quickly compute the value of $\tilde{\sigma}(\boldsymbol{s})$ based on arithmetic functions related to the bit length.

Definition 12: For a vector $\boldsymbol{s} = [s_1, s_2, \ldots, s_n] \in \mathbb{N}_0^n$, we define $L_{max}(\boldsymbol{s}) = \max\{L(s_i) \mid 1 \leq i \leq n\}$, which is the maximum bit length of s_i in $\boldsymbol{s}$. $\#_d(\boldsymbol{s})$ is defined by the number of elements s_i whose bit length is d or more:

$$\#_d(\boldsymbol{s}) = |\{i \mid 1 \leq i \leq n,\ L(s_i) \geq d\}|.$$

□

Note that $\#_0(\boldsymbol{s}) = n$ and $\#_d(\boldsymbol{s}) = 0$ for $d > L_{max}(\boldsymbol{s})$.

Definition 13: For a vector $\boldsymbol{s} \in \mathbb{N}_0^n$, $\kappa(\boldsymbol{s})$ is defined as follows.

$$\kappa_d(\boldsymbol{s}) = (d-1) + \#_d(\boldsymbol{s})$$

$$\kappa(\boldsymbol{s}) = \begin{cases} 0 & (\boldsymbol{s} = \boldsymbol{0}) \\ \max\{\kappa_d(\boldsymbol{s}) \mid 1 \leq d \leq L_{max}(\boldsymbol{s})\} & (\text{otherwise}) \end{cases}$$

□

Example 7: Let $\boldsymbol{s} = [1, 4, 8, 9]$. $\kappa_1(\boldsymbol{s}) = (1-1) + \#_1(\boldsymbol{s}) = 0 + 4 = 4$, $\kappa_2(\boldsymbol{s}) = (2-1) + \#_2(\boldsymbol{s}) = 1 + 3 = 4$, $\kappa_3(\boldsymbol{s}) = (3-1)+\#_3(\boldsymbol{s}) = 2+3 = 5$, $\kappa_4(\boldsymbol{s}) = (4-1)+\#_4(\boldsymbol{s}) = 3+2 = 5$. Then, $\kappa(\boldsymbol{s}) = 5$. □

κ defined above is a faster version of $\tilde{\sigma}$. Their computational processes look quite different. Nevertheless, both $\kappa(\boldsymbol{s})$ and $\tilde{\sigma}(\boldsymbol{s})$ produce the same value for an arbitrary vector $\boldsymbol{s} \in \mathbb{N}_0^n$. In the rest of this section, $\kappa(\boldsymbol{s}) = \tilde{\sigma}(\boldsymbol{s})$ will be proved and then an algorithm for $\kappa(\boldsymbol{s})$ with a complexity of $O(n)$ will be presented.

The definition of κ is more complex than a plain evaluation of the bit length of the elements in $\boldsymbol{s}$. This is because $\tilde{\boldsymbol{\Phi}}$ used in $\tilde{\sigma}$ has cases other than the right-shift operation. In order to conclude that $\kappa(\boldsymbol{s}) = \tilde{\sigma}(\boldsymbol{s})$, we need to analyze all the cases of $\tilde{\boldsymbol{\Phi}}$ carefully. The analysis will be made in the proof of Lemma 11, by utilizing Lemmas 9 and 10.

Lemma 9: Suppose that $\boldsymbol{s}$ and $\boldsymbol{s}'$ are vectors in $\mathbb{N}_0^n$, d and d' are positive integers, c is a constant, and '$\asymp$' is one of the following relations: '$=$', '$\leq$', '$<$', '$\geq$', and '$>$'. If $\#_d(\boldsymbol{s}) \asymp \#_{d'}(\boldsymbol{s}') + c$, then $\kappa_d(\boldsymbol{s}) \asymp \kappa_{d'}(\boldsymbol{s}') + d - d' + c$. □

Proof. By the definition of κ_d, $\kappa_d(\boldsymbol{s}) = (d-1) + \#_d(\boldsymbol{s}) \asymp (d-1) + \#_{d'}(\boldsymbol{s}') + c = (d'-1) + \#_{d'}(\boldsymbol{s}') + d - d' + c = \kappa_{d'}(\boldsymbol{s}') + d - d' + c$. □

Lemma 10: For a vector $\boldsymbol{s} \in \mathbb{N}_0^n \setminus \{\boldsymbol{0}\}$ and an integer d with $d \geq 2$, $\kappa_d(\tilde{\boldsymbol{\Phi}}(\boldsymbol{s})) = \kappa_{d+1}(\boldsymbol{s}) - 1$. □

Proof. Suppose that $[s_1, s_2, \ldots, s_n] = \boldsymbol{s}$ and $\boldsymbol{s}' = \tilde{\boldsymbol{\Phi}}(\boldsymbol{s}) = [s'_1, s'_2, \ldots, s'_n]$. By the definition of $\tilde{\boldsymbol{\Phi}}$, we have the following equation for $L(s'_i)$ and $L(s_i)$.

$$L(s'_i) = \begin{cases} 0 & (s_i = 0) \\ 0 & (s_i = 1 \text{ and } i = \iota(\boldsymbol{s})) \\ 1 & (s_i = 1 \text{ and } i \neq \iota(\boldsymbol{s})) \\ L(s_i) - 1 & (s_i \geq 2) \end{cases} \tag{3}$$

From Equation (3) and the assumption of the lemma $d \geq 2$, if $L(s_i) \geq d+1$, then $L(s'_i) = L(s_i) - 1$ and $L(s'_i) \geq d$. Conversely, if $L(s'_i) \geq d$, then $L(s'_i) = L(s_i) - 1$ and $L(s_i) \geq d+1$. It follows that $L(s'_i) \geq d$ holds if and only if $L(s_i) \geq d+1$. Hence, $\#_d(\boldsymbol{s}') = \#_{d+1}(\boldsymbol{s})$, and therefore $\kappa_d(\boldsymbol{s}') = \kappa_{d+1}(\boldsymbol{s}) - 1$ by Lemma 9. □

Lemma 11: For a vector $\boldsymbol{s} \in \mathbb{N}_0^n \setminus \{\boldsymbol{0}\}$, $\kappa(\tilde{\boldsymbol{\Phi}}(\boldsymbol{s})) = \kappa(\boldsymbol{s}) - 1$. □

Proof. Suppose that $[s_1, s_2, \ldots, s_n] = \boldsymbol{s}$ and $\boldsymbol{s}' = \tilde{\boldsymbol{\Phi}}(\boldsymbol{s}) = [s'_1, s'_2, \ldots, s'_n]$. Then, we have Equation (3) for $L(s'_i)$ and $L(s_i)$ as we have seen in the proof of Lemma 10. Since $\boldsymbol{s} \neq \boldsymbol{0}$, $L_{max}(\boldsymbol{s}') = L_{max}(\boldsymbol{s}) - 1$ holds from Equation (3).
(Case of $\iota(\boldsymbol{s}) = 0$): Here, no elements s_i equal 1. Hence, $\#_1(\boldsymbol{s}) = \#_2(\boldsymbol{s})$, and therefore $\kappa_1(\boldsymbol{s}) = \kappa_2(\boldsymbol{s}) - 1$ by Lemma 9. Since $\kappa_1(\boldsymbol{s}) = \kappa_2(\boldsymbol{s}) - 1 < \kappa_2(\boldsymbol{s})$, we can exclude $\kappa_1(\boldsymbol{s})$ from the candidates for $\kappa(\boldsymbol{s})$:

$$\kappa(\boldsymbol{s}) = \max\{\kappa_d(\boldsymbol{s}) \mid 2 \leq d \leq L_{max}(\boldsymbol{s})\}. \tag{4}$$

Next, consider $\kappa_2(\boldsymbol{s})$. The absence of elements $s_i = 1$ simplifies Equation (3) as follows.

$$L(s'_i) = \begin{cases} 0 & (s_i = 0) \\ L(s_i) - 1 & (s_i \geq 2) \end{cases}$$

This equation shows that $L(s'_i) \geq 1$ holds if and only if $L(s_i) \geq 2$. Hence, $\#_1(\boldsymbol{s}') = \#_2(\boldsymbol{s})$, and therefore $\kappa_1(\boldsymbol{s}') = \kappa_2(\boldsymbol{s}) - 1$ by Lemma 9. For $\kappa_3(\boldsymbol{s}), \kappa_4(\boldsymbol{s})$, and so on, $\kappa_d(\boldsymbol{s}') = \kappa_{d+1}(\boldsymbol{s}) - 1$ holds by Lemma 10. Combining these equations with the definition of κ, we have $\kappa(\boldsymbol{s}') = \max\{\kappa_d(\boldsymbol{s}') \mid 1 \leq d \leq L_{max}(\boldsymbol{s}')\} = \max\{\kappa_{d+1}(\boldsymbol{s}) - 1 \mid 1 \leq d \leq L_{max}(\boldsymbol{s}')\} = \max\{\kappa_d(\boldsymbol{s}) - 1 \mid 2 \leq d \leq L_{max}(\boldsymbol{s}') + 1 (= L_{max}(\boldsymbol{s}))\} = \max\{\kappa_d(\boldsymbol{s}) \mid 2 \leq d \leq L_{max}(\boldsymbol{s})\} - 1$. From this equation and Equation (4), we conclude $\kappa(\boldsymbol{s}') = \kappa(\boldsymbol{s}) - 1$.
(Case of $\iota(\boldsymbol{s}) \neq 0$): In this case, $L(s'_{\iota(\boldsymbol{s})}) = 0$ and $L(s_{\iota(\boldsymbol{s})}) = 1$ from Equation (3). For other indices $i \neq \iota(\boldsymbol{s})$, $L(s'_i) \geq 1$ holds if and only if $L(s_i) \geq 1$. Hence, $\#_1(\boldsymbol{s}') = \#_1(\boldsymbol{s}) - 1$ holds, and then, Lemma 9 yields

$$\kappa_1(\boldsymbol{s}') = \kappa_1(\boldsymbol{s}) - 1. \tag{5}$$

Next, consider $\kappa_2(\boldsymbol{s})$. Equation (3) shows that if $L(s_i) \geq 2$, then $L(s'_i) \geq 1$, but its converse does not hold. Hence, $\#_2(\boldsymbol{s}) \leq \#_1(\boldsymbol{s}')$, and therefore $\kappa_2(\boldsymbol{s}) \leq \kappa_1(\boldsymbol{s}')+1$ by Lemma 9. Since $\kappa_1(\boldsymbol{s}')+1 = \kappa_1(\boldsymbol{s})$ from Equation (5), we have $\kappa_2(\boldsymbol{s}) \leq \kappa_1(\boldsymbol{s}') + 1 = \kappa_1(\boldsymbol{s})$. Since $\kappa_2(\boldsymbol{s}) \leq \kappa_1(\boldsymbol{s})$, we can exclude $\kappa_2(\boldsymbol{s})$ from the candidates for $\kappa(\boldsymbol{s})$:

$$\kappa(\boldsymbol{s}) = \max\{\kappa_d(\boldsymbol{s}) \mid 1 \leq d \leq L_{max}(\boldsymbol{s}),\ d \neq 2\}. \tag{6}$$

For $\kappa_3(\boldsymbol{s}), \kappa_4(\boldsymbol{s})$, and so on, $\kappa_d(\boldsymbol{s}') = \kappa_{d+1}(\boldsymbol{s}) - 1$ holds by Lemma 10. Combining this equation, Equation (5), and the definition of κ, we have $\kappa(\boldsymbol{s}') = \max\{\kappa_d(\boldsymbol{s}') \mid 1 \leq d \leq L_{max}(\boldsymbol{s}')\} = \max\{\kappa_d(\boldsymbol{s}) - 1 \mid 1 \leq d \leq L_{max}(\boldsymbol{s}') + 1 (= L_{max}(\boldsymbol{s})),\ d \neq 2\} = \max\{\kappa_d(\boldsymbol{s}) \mid 1 \leq d \leq L_{max}(\boldsymbol{s}),\ d \neq 2\} - 1$. From this equation and Equation (6), we conclude $\kappa(\boldsymbol{s}') = \kappa(\boldsymbol{s}) - 1$. □

Theorem 6: For a vector $\boldsymbol{s} \in \mathbb{N}_0^n$,

$$\kappa(\boldsymbol{s}) = \tilde{\sigma}(\boldsymbol{s})$$

holds. □

Proof. The proof is by the mathematical induction on $\kappa(\boldsymbol{s})$. If $\kappa(\boldsymbol{s}) = 0$, then $\boldsymbol{s} = \boldsymbol{0}$, and therefore $\tilde{\sigma}(\boldsymbol{s}) = 0$. Thus, the theorem holds for $\kappa(\boldsymbol{s}) = 0$.

Assume that the theorem holds for any vector $\boldsymbol{s}'$ such that $\kappa(\boldsymbol{s}') = k$, i.e., $\kappa(\boldsymbol{s}') = \tilde{\sigma}(\boldsymbol{s}') = k$. Using this assumption, we prove the theorem for $\kappa(\boldsymbol{s}) = k + 1$. Suppose that $\boldsymbol{s}$ is a vector such that $\kappa(\boldsymbol{s}) = k + 1$. Then, since $\kappa(\boldsymbol{s}) = \kappa(\tilde{\boldsymbol{\Phi}}(\boldsymbol{s})) + 1$ by Lemma 11, $\kappa(\tilde{\boldsymbol{\Phi}}(\boldsymbol{s})) = k$ holds. By using the induction hypothesis, $\kappa(\tilde{\boldsymbol{\Phi}}(\boldsymbol{s})) = \tilde{\sigma}(\tilde{\boldsymbol{\Phi}}(\boldsymbol{s}))$. By Proposition 4, $\tilde{\sigma}(\tilde{\boldsymbol{\Phi}}(\boldsymbol{s})) = \tilde{\sigma}(\boldsymbol{s}) - 1$. From these equations, we have $\kappa(\boldsymbol{s}) = \kappa(\tilde{\boldsymbol{\Phi}}(\boldsymbol{s})) + 1 = \tilde{\sigma}(\tilde{\boldsymbol{\Phi}}(\boldsymbol{s})) + 1 = \tilde{\sigma}(\boldsymbol{s}) - 1 + 1 = \tilde{\sigma}(\boldsymbol{s})$. Thus, the inductive step was proved. □

Example 8: By Examples 3 and 7, $\kappa([1,4,8,9]) = \tilde{\sigma}([1,4,8,9]) = 5$. □

Theorem 6 shows that the value of $\tilde{\sigma}(\boldsymbol{s})$ can also be obtained by calculating $\kappa(\boldsymbol{s})$. For computational efficiency, we use $\kappa(\boldsymbol{s})$ mainly instead of $\tilde{\sigma}(\boldsymbol{s})$ in the rest of the paper although $\tilde{\sigma}(\boldsymbol{s})$ provided theoretical insights for the comparison with σ-$lb(\boldsymbol{s})$ in Sects. 3 and 4.

```
1:  function Kappa(s)
2:                                  ▷ Input: s is a vector in N_0^n
3:                                  ▷ Output: the value of κ(s)
4:      Var b, h ← 0 : Vector
5:      Var l, l*, k, k* ← 0 : Integer
6:      if s = 0 then retrun 0
7:      for i ← 1 to n do
8:          l ← L(s[i])
9:          if l ≠ 0 then
10:             b[l] ← b[l] + 1
11:             if l* < l then l* ← l
12:     h[n] ← b[n]
13:     for i ← n − 1 downto 1 do
14:         h[i] ← b[i] + h[i + 1]
15:     for i ← 1 to l* do
16:         k ← h[i] + (i − 1)
17:         if k* < k then k* ← k
18:     return k*
```

Fig. 3 Kappa: a fast algorithm for κ

Table 1 Average computation time [μs] per vector

n	Kappa	SigmaLB3
3	0.052	4.5
4	0.058	10.0
5	0.065	27.1
6	0.070	54.8
7	0.078	94.4
8	0.083	149.9
9	0.091	240.1
10	0.097	374.7

We now present an algorithm for $\kappa(\boldsymbol{s})$ as Kappa in Fig. 3, in which vectors are expressed as 1-indexed arrays. In Kappa, the calculation of $\#_d(\boldsymbol{s})$ is improved by a technique similar to bucket sorting under the assumption that $L(s_i) \leq n$ for any s_i. Since Kappa is customized for the characteristic vector $\boldsymbol{\Lambda}(F)$ of a reversible function F, the bit length of s_i is assumed to be at most n by Proposition 1. The **for** loop starting at Line 7 counts the frequency of the bit length of the elements in $\boldsymbol{s}$ except those with a bit length of 0, storing the results in $\boldsymbol{b}$. The **for** loop starting at Line 13 calculates the suffix sums of $\boldsymbol{b}$, resulting in $\boldsymbol{h} = [\#_1(\boldsymbol{s}), \#_2(\boldsymbol{s}), \ldots, \#_n(\boldsymbol{s})]$. The **for** loop at Line 15 identifies the maximum value of $\kappa_i(\boldsymbol{s})$ for $1 \leq i \leq n$, which is finally returned at Line 18.

The time complexity of Kappa is $O(n)$ if L is assumed to be a constant-time operation. The assumption is based on the fact that L, i.e., the bit length of an integer, is mathematically a simple logarithm as in Definition 11, and is practically a built-in operation in many programming languages such as Mathematica and Common Lisp. Note that the evaluation of the complexity may vary according to the presumed complexity of the L operation because Kappa invokes this operation n times.

6. Experimental Results

To measure the computation time, we implemented Kappa in Common Lisp (SBCL), and calculated the values of $\kappa(\boldsymbol{\Lambda}(F))$ for all 3-variable functions (40,320 in total) as well as

Table 2 Average of lower bounds

n	$\kappa(\boldsymbol{\Lambda}(F))$	σ-$lb(\boldsymbol{\Lambda}(F))$
3	3.98	4.09
4	6.00	6.08
5	8.03	8.05
6	10.02	10.03
7	12.01	12.01
8	14.00	14.01
9	16.00	16.00
10	18.00	18.00

for 50,000 randomly-generated n-variable functions ranging from $n = 4$ to 10. The program was executed on a computer with Ubuntu 22.04LTS / Core i9-12900K CPU (3.2GHz). Table 1 shows the average computation time in microseconds per characteristic vector. κ is much faster than σ-lb even though the fastest algorithm named SigmaLB3 [25] for calculating σ-lb was used as the implementation of σ-lb. It should be mentioned that SigmaLB3 consumes a lot of memory for caching to reduce its computation time. Without this caching strategy, it has been reported [25] that the computation time for σ-lb grows exponentially with n. Regarding memory usage, Kappa apparently works with only $O(n)$ memory.

Kappa is very fast, but the growth of the experimental times in Table 1 does not look $O(n)$. A considerable factor behind this is the overhead necessary for Kappa to work as a procedure. We measured the time of this overhead, which includes the function call for Kappa and the allocation of local variables and arrays. The overhead was about 0.032[μs]. When we adjust the times of Kappa in Table 1 by subtracting this overhead, the resultant times become approximately proportional to n. The calculation of Kappa is so simple that the necessary overhead seems to be a major part of the total computation time. From this observation, we can conclude that Kappa is sufficiently quick in practice.

By Theorem 3, $\kappa(\boldsymbol{\Lambda}(F))$ is a lower bound on the GC of a circuit of a reversible function F. As an experimental comparison between $\kappa(\boldsymbol{\Lambda}(F))$ and σ-$lb(\boldsymbol{\Lambda}(F))$, we computed both lower bounds for all (40,320) 3-variable functions and 50,000 randomly-generated n-variable functions ranging from $n = 4$ to 10 as well as the experiments in Table 1. The average values of the lower bounds for these functions are given in Table 2. The two lower bounds become closer to each other with increase in the number of variables. On average, the difference in quality between κ and σ-lb is very slight. In this paper, we omit a comparison with the minimum circuits or the upper bounds. A detailed comparison between those data and σ-$lb(\boldsymbol{\Lambda}(F))$ has been made in the work [25].

Table 3 shows the match-to-mismatch ratio of $\kappa(\boldsymbol{\Lambda}(F))$ and σ-$lb(\boldsymbol{\Lambda}(F))$ for functions. Theorem 5 states that there can be two cases: either $\kappa(\boldsymbol{\Lambda}(F)) = \sigma$-$lb(\boldsymbol{\Lambda}(F))$ or $\kappa(\boldsymbol{\Lambda}(F)) = \sigma$-$lb(\boldsymbol{\Lambda}(F)) - 1$. In Table 3, more than 90% of the functions for $n \geq 4$ fall into the former case, and the percentage of this case increases with n. Experimentally, the two lower bounds are equal for most functions. One instance of a mismatch in $n = 4$ is the function $F = [6, 12, 14, 11, 3, 4, 13, 1, 10, 15, 7, 2, 8, 9, 5, 0]$ in permutation notation. Its characteristic vector is $\boldsymbol{\Lambda}(F) = [9, 9, 11, 7]$, and its lower bounds are $\kappa(\boldsymbol{\Lambda}(F)) = 6$ and σ-$lb(\boldsymbol{\Lambda}(F)) = 7$. Another such instance is the function $F = [10, 15, 6, 13, 9, 2, 7, 5, 3, 8, 0, 14, 12, 4, 1, 11]$ with $\boldsymbol{\Lambda}(F) = [6, 10, 11, 10]$, $\tilde{\boldsymbol{\Phi}}(\boldsymbol{\Lambda}(F)) = [3, 5, 5, 5]$, $\kappa(\boldsymbol{\Lambda}(F)) = 6$, and σ-$lb(\boldsymbol{\Lambda}(F)) = 7$. Like these examples, if $\boldsymbol{\Lambda}(F)$ or $\tilde{\boldsymbol{\Phi}}(\boldsymbol{\Lambda}(F))$ contains many odd numbers, the two lower bounds may differ due to the discrepancy between the ways of halving in $\boldsymbol{\Phi}$ and $\tilde{\boldsymbol{\Phi}}$ operations.

7. Conclusion

We have introduced a halving operation $\tilde{\boldsymbol{\Phi}}$ for the characteristic vectors of reversible functions. Through a careful analysis of $\tilde{\boldsymbol{\Phi}}$, we proposed a new lower bound $\tilde{\sigma}$ on the GC of reversible circuits that realize a given reversible function. To improve computational efficiency, a faster version of $\tilde{\sigma}$, called κ, was presented. The theory and experiments have confirmed that the value of κ is almost the same as that of the previous lower bound σ-lb. κ can be calculated much faster than σ-lb by several orders of magnitude. The complexity of our proposed algorithm for κ is $O(n)$ with respect to both time and space. Our future work includes refining our bounds to take into account practical cost metrics such as the number of controls and applying our bounds to the reversible logic synthesis.

References

[1] T. Hirayama, R. Suzuki, K. Yamanaka, and Y. Nishitani, "Quick computation of the lower bound on the gate count of toffoli-based reversible logic circuits," Proc. 53rd ISMVL, Matsue, Japan, pp.153–157, May 2023.

[2] R. Wille and R. Drechsler, Towards a Design Flow for Reversible Logic, Springer, 2010.

[3] P. Gupta, A. Agrawal, and N.K. Jha, "An algorithm for synthesis of reversible logic circuits," IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems, vol.25, no.11, pp.2317–2330, Nov. 2006.

[4] K. Iwama, Y. Kambayashi, and S. Yamashita, "Transformation rules for designing CNOT-based quantum circuits," Proc. 39th Design Automation Conference, USA, pp.419–424, 2002.

[5] D. Maslov, G. Dueck, and D. Miller, "Toffoli network synthesis with templates," IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems, vol.24, no.6, pp.807–817, June 2005.

[6] D. Miller and G. Dueck, "Spectral techniques for reversible logic synthesis," Proc. 6th Int. Symp. Representations and Methodology of Future Computing Technologies, Trier, Germany, pp.56–62, March 2003.

[7] G. Dueck, D. Maslov, and D. Miller, "Transformation-based synthesis of networks of Toffoli/Fredkin gates," Proc. Canadian Conference on Electrical and Computer Engineering, pp.211–214, vol.1, May 2003.

[8] P. Kerntopf, "A new heuristic algorithm for reversible logic synthesis," Proc. 41st Design Automation Conference, pp.834–837, 2004.

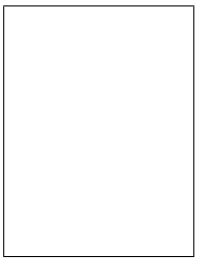
[9] D. Maslov, G. Dueck, and D. Miller, "Synthesis of Fredkin-Toffoli reversible networks," IEEE Trans. VLSI Syst., vol.13, no.6, pp.765–769, June 2005.

[10] H. Thapliyal and N. Ranganathan, "Design of reversible sequential circuits optimizing quantum cost, delay and garbage outputs," ACM

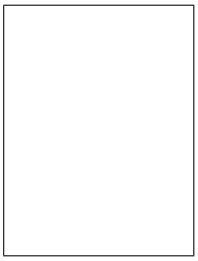
Table 3 Match-to-mismatch ratio of two lower bounds for functions.

Cases	$n = 3$	$n = 4$	$n = 5$	$n = 6$	$n = 7$	$n = 8$	$n = 9$	$n = 10$
$\kappa(\boldsymbol{\Lambda}(F)) = \sigma\text{-}lb(\boldsymbol{\Lambda}(F))$	35,952 (89.2%)	92.0%	97.8%	99.1%	99.5%	99.8%	99.9%	99.9%
$\kappa(\boldsymbol{\Lambda}(F)) = \sigma\text{-}lb(\boldsymbol{\Lambda}(F)) - 1$	4,368 (10.8%)	8.0%	2.2%	0.9%	0.5%	0.2%	0.1%	0.1%
Total	40,320 (100%)	100%	100%	100%	100%	100%	100%	100%

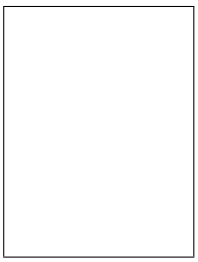
Journal of Emerging Technologies in Computing Systems, vol.6, no.4, article 14, Dec. 2010.
[11] H. Thapliyal and N. Ranganathan, "Design of efficient reversible logic based binary and BCD adder circuits," ACM Journal of Emerging Technologies in Computing Systems, vol.9, no.3, pp.17:1–17:31, Oct. 2013.
[12] M. Saeedi and I.L. Markov, "Synthesis and optimization of reversible circuits – a survey," ACM Computing Surveys, vol.45, no.2, article 21, Feb. 2013.
[13] K.N. Patel, I.L. Markov, and J.P. Hayes, "Optimal synthesis of linear reversible circuits," Quantum Information and Computation, vol.8, no.3, pp.282–294, March 2008.
[14] N. Abdessaied, M. Amy, R. Drechsler, and M. Soeken, "Complexity of reversible circuits and their quantum implementations," Theoretical Computer Science, vol.618, pp.85–106, 2016.
[15] D. Maslov and G. Dueck, "Reversible cascades with minimal garbage," IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems, vol.23, no.11, pp.1497–1509, Nov. 2004.
[16] V.V. Shende, A.K. Prasad, I.L. Markov, and J.P. Hayes, "Synthesis of reversible logic circuits," IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems, vol.22, no.6, pp.710–722, June 2003.
[17] M. Soeken, N. Abdessaied, and R. Drechsler, "A framework for reversible circuit complexity," Proc. International Workshop on Boolean Problems 2014, arXiv:1407.5878, Aug. 2014.
[18] D.V. Zakablukov, "On asymptotic gate complexity and depth of reversible circuits without additional memory," arXiv:1504.06876, Feb. 2016.
[19] S. Popescu, B. Groisman, and S. Massar, "Lower bound on the number of Toffoli gates in a classical reversible circuit through quantum information concepts," Physical Review Letters, vol.95, no.12, pp.120503-1–120503-4, Sept. 2005.
[20] D. Maslov, "Optimal and asymptotically optimal NCT reversible circuits by the gate types," Quantum Information & Computation, vol.16, no.13&14, pp.1096–1112, Oct. 2016.
[21] M. Saeedi, M.S. Zamani, M. Sedighi, and Z. Sasanian, "Reversible circuit synthesis using a cycle-based approach," ACM Journal on Emerging Technologies in Computing Systems, vol.6, no.4, article 13, Dec. 2010.
[22] R. Wille and R. Drechsler, "Synthesis of Boolean functions in reversible logic," in Progress in Applications of Boolean Functions, ed. T. Sasao and J.T. Butler, ch. 4, pp.79–96, Morgan & Claypool Publishers, 2010.
[23] N.M. Nayeem and J.E. Rice, "Improved ESOP-based synthesis of reversible logic," Proc. Reed-Muller 2011 Workshop, Tuusula, Finland, pp.57–62, May 2011.
[24] M. Higashiohno, T. Hirayama, and Y. Nishitani, "A lower bound on the number of Toffoli gates in reversible logic circuits," IEICE Trans. Fundamentals of Electronics, Communications and Computer Sciences, vol.J92-A, no.4, pp.263–266, April 2009.
[25] T. Hirayama, H. Sugawara, K. Yamanaka, and Y. Nishitani, "A lower bound on the gate count of Toffoli-based reversible logic circuits," IEICE Trans. Information and Systems, vol.E97-D, no.9, pp.2253–2261, Sept. 2014.
[26] M. Szyprowski and P. Kerntopf, "Estimating the quality of complexity measures in heuristics for reversible logic synthesis," Proc. IEEE Congress on Evolutionary Computation (CEC) 2010, pp.1–8, July 2010.
[27] X. Chen and M.L. Bushnell, Efficient Branch and Bound Search with Application to Computer-Aided Design, Springer, 1996.
[28] M. Davio, J.P. Deschamps, and A. Thayse, Discrete and Switching Functions, McGraw-Hill International, 1978.
[29] T. Sasao, Switching Theory For Logic Synthesis, Kluwer Academic Publishers, Feb. 1999.



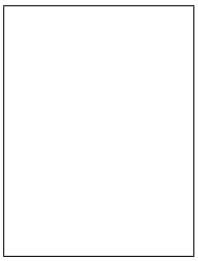
Takashi Hirayama received his B.E., M.E., and Ph.D. degrees in computer science from Gunma University in 1994, 1996, and 1999, respectively. From 1999 to 2001 he was a research assistant at Ashikaga Institute of Technology. He is currently an associate professor of Faculty of Science and Engineering, Iwate University. His research interests include high-level and logic synthesis, logic optimization algorithms, and design for testability for VLSIs.



Rin Suzuki received his B.E. and M.E degrees in electrical engineering and computer science from Iwate University, Morioka, Japan, in 2016 and 2018, respectively. His research interests include reversible logic synthesis and optimization algorithms.



Katsuhisa Yamanaka is a professor of Faculty of Science and Engineering, Iwate University. He received his B.E., M.E., and Ph.D. degrees in computer science from Gunma University in 2003, 2005, and 2007, respectively. His research interests include combinatorial algorithms and graph algorithms.



Yasuaki Nishitani received his B.E. degree in electrical engineering, M.E. and Ph.D. degrees in computer science from Tohoku University in 1975, 1977, and 1984, respectively. In 1981 he joined the Software Product Engineering Laboratory at the NEC Corporation. From 1987 to 2000 he was an associate professor in the Department of Computer Science, Gunma University. From 2000 to 2017, he was a professor of Faculty of Science and Engineering, Iwate University. His current research interests include switching theory, software engineering, and distributed algorithms.