Tight Bounds on the Synthesis of 3-bit Reversible Circuits: NFT Library

Ahmed Younes*
Department of Mathematics and Computer Science
Faculty of Science, Alexandria University
Alexandria, Egypt
October 1, 2018

Abstract

The reversible circuit synthesis problem can be reduced to permutation group. This allows Schreier-Sims Algorithm for the strong generating set-finding problem to be used to find tight bounds on the synthesis of 3-bit reversible circuits using the NFT library. The tight bounds include the maximum and minimum length of 3-bit reversible circuits, the maximum and minimum cost of 3-bit reversible circuits. The analysis shows better results than that found in the literature for the lower bound of the cost. The analysis also shows that there are 1960 universal reversible sub-libraries from the main NFT library.

Keywords: Reversible circuit; Quantum Cost; Circuit Optimization; Group Theory.

1 Introduction

Reversible logic \[2, 11\] is one of the hot areas of research. It has many applications in quantum computation \[13, 23\], low-power CMOS \[9, 5\] and many more. Synthesis of reversible circuits cannot be done using conventional ways \[29\]. Synthesis and optimization of Boolean systems on non-standard computers that promise to do computation more powerfully \[26\] than classical computers, such as quantum computers, is an essential aim in the exploration of the benefits that may be gain from such systems.

A lot of work has been done trying to find an efficient reversible circuit for an arbitrary reversible function. In one of the research directions, it was shown that the process of synthesizing linear reversible circuits can be reduced to a row reduction problem of \(n \times n\) non-singular matrix \[24\]. Standard row reduction methods such as Gaussian elimination and LU-decomposition have been proposed \[3\]. In another research direction, search algorithms and template matching tools using reversible gates libraries have been used \[10, 17, 21, 22\]. These will work efficiently for small circuits. A method is given in \[15\], where a very useful set of transformations for Boolean quantum circuits is shown. In this method, extra auxiliary bits are used in the construction that will increase the hardware cost. In \[31\], it was shown that there is a direct correspondence between reversible Boolean operations and certain forms of classical logic known as Reed-Muller expansions. This arises the possibility of handling the problem of synthesis and optimization of reversible Boolean logic within the field of Reed-Muller logic. A lot of work has been done trying to find an efficient reversible circuit for an arbitrary multi-output Boolean functions by using templates \[18, 19\] and data-structure-based optimization \[25\]. A method to generate an optimal 4-bit reversible circuits has been proposed \[13\]. Benchmarks for reversible circuits have been established \[20\].

Recently, the study of reversible logic synthesis problem using group theory is gaining more attention. Investigation on the universality of the basic building blocks of reversible circuit has been done \[28, 6\]. A relation between Young subgroups and the reversible logic synthesis problem has been proposed \[7\]. A comparison between the decomposition of reversible circuit and quantum circuit using group theory has been shown \[8\]. A GAP-based algorithms to synthesize reversible circuits for various types of gate with various gate costs has been proposed \[30\].

The aim of the paper is to answer the following questions for the synthesis of 3-bit reversible circuits using NFT library:

*ayounes2@yahoo.com or ayounes@alexu.edu.eg
1. What are the maximum and minimum lengths of reversible circuits for a reversible function? What are the costs of these circuits?

2. What are the maximum and minimum costs of reversible circuits for a reversible function? What are the length of these circuits?

3. What are the upper-bound and lower-bound on lengths of reversible circuits for all 3-bit reversible functions using NFT library? What are the costs of these bounds?

4. What are the upper-bound and lower-bound on costs of reversible circuits for all 3-bit reversible functions using NFT library? What are the lengths of these bounds?

5. Is there any sub-library of NFT that can act as a universal reversible gate library? If so, what is the best universal reversible gate library that gives the best length, worst length, best cost and worst cost? How many sub-libraries that can act as a universal reversible gate library?

The paper is organized as follows: Section 2 gives a short background on the synthesis of reversible circuit problem and shows the reduction the problem to permutation group. Section 3 shows Schreier-Sims Algorithm for the strong generating set-finding problem that is used to calculate the bounds on the synthesis of 3-bit reversible circuits problem using NFT library. Section 4 shows the results of the experiments. The paper ends up with a summary and conclusion in Section 5.

2 Background

2.1 Reversible Circuits

To build a reversible circuit with \( n \) variables, an \( n \times n \) reversible circuit is used. \( C^n \) NOT gate is the main primitive gate that is used to build the circuit since it is shown to be universal for reversible computation \[29\]. \( C^n \) NOT gate is defined as follows:

**Definition 2.1 (\( C^n \) NOT gate)**

\( C^n \) NOT is a reversible gate denoted as,

\[
C^n \text{NOT}(x_1, x_2, \ldots, x_{n-1}; f),
\]

with \( n \) inputs: \( x_1, x_2, \ldots, x_{n-1} \) (known as control bits) and \( f_{in} \) (known as target bit), and \( n \) outputs: \( y_1, y_2, \ldots, y_{n-1} \) and \( f_{out} \). The operation of the \( C^n \) NOT gate is defined as follows,

\[
y_i = x_i, \text{ for } 1 \leq i \leq n-1,
\]

\[
f_{out} = f_{in} \oplus x_1x_2\ldots x_{n-1},
\]

i.e. the target bit will be flipped if and only if all the control bits are set to 1. Some special cases of the general \( C^n \) NOT gate have their own names, \( C^1 \) NOT gate with no control bits is called \( \text{NOT} \) gate as shown in Fig. 1-b, where the bit will be flipped unconditionally. \( C^2 \) NOT gate with one control bit is called \( \text{Feynman} \) gate as shown in Fig. 1-c. \( C^3 \) NOT gate with two control bits is called \( \text{Toffoli} \) gate as shown in Fig. 1-d.

![Figure 1: \( C^n \) NOT gates. The black circle • indicates the control bits, and the symbol \( \oplus \) indicates the target bit. (a)\( C^n \) NOT gate with \( n - 1 \) control bits. (b) \( C^1 \) NOT gate with no control bits. (c)\( C^2 \) NOT gate with one control bit. (d)\( C^3 \) NOT gate with two control bits.](image-url)
For the sake of readability $C^1 \text{NOT}$, $C^2 \text{NOT}$ and $C^3 \text{NOT}$ will be written for short as $N$, $F$ and $T$ respectively where the control and/or target bits will be shown in the subscript of the gate, for example, a 3-bit reversible circuit synthesized using $N,F$ and $T$ gates can contain 12 possible gates, as shown in Fig.2 that perform as follows:

\[
\begin{align*}
N_1 &: (x_1, x_2, x_3) \rightarrow (x_1 \oplus 1, x_2, x_3) \\
N_2 &: (x_1, x_2, x_3) \rightarrow (x_1, x_2 \oplus 1, x_3) \\
N_3 &: (x_1, x_2, x_3) \rightarrow (x_1, x_2, x_3 \oplus 1) \\
F_{12} &: (x_1, x_2, x_3) \rightarrow (x_1, x_2 \oplus x_1, x_3) \\
F_{13} &: (x_1, x_2, x_3) \rightarrow (x_1, x_2, x_3 \oplus x_1) \\
F_{23} &: (x_1, x_2, x_3) \rightarrow (x_1, x_2, x_3 \oplus x_2) \\
F_{21} &: (x_1, x_2, x_3) \rightarrow (x_1 \oplus x_2, x_2, x_3) \\
F_{31} &: (x_1, x_2, x_3) \rightarrow (x_1 \oplus x_3, x_2, x_3) \\
T_{123} &: (x_1, x_2, x_3) \rightarrow (x_1, x_2, x_3 \oplus x_1 x_2) \\
T_{132} &: (x_1, x_2, x_3) \rightarrow (x_1, x_2 \oplus x_1 x_3, x_3) \\
T_{321} &: (x_1, x_2, x_3) \rightarrow (x_1 \oplus x_2 x_3, x_2, x_3)
\end{align*}
\]

A reversible circuit is a cascade of one or more $C^n \text{NOT}$ gates, for example, a 3-bit reversible circuit shown in Fig.4.a can be written for short as $(T_{123} F_{23} N_1 T_{132} F_{13} F_{31})$, where the truth table of this circuit is shown in Fig.4.b. Another way to write this truth table is $(0, 1, 2, 3, 4, 5, 6, 7) \rightarrow (2, 6, 5, 4, 7, 1, 0, 3)$, or simply $(2, 6, 5, 4, 7, 1, 0, 3)$ which is called the specification of the circuit.

### 2.2 Quantum Cost

*Quantum cost* is a term used to refer to the technological cost of building $C^n \text{NOT}$ gates. The quantum cost of a reversible circuit is subject to optimization as well as the number of $C^n \text{NOT}$ gates used in the circuit. The quantum cost of a $C^n \text{NOT}$ gate is based primarily on the number of bits involved in the gate, i.e. the number of elementary operations required to build the $C^n \text{NOT}$ gate [1].

\[
\begin{align*}
\text{Figure 3: Decomposition of } T \text{ gate as 5 elementary 2-qubit gates.}
\end{align*}
\]

When implementing a reversible gate circuit on a quantum computer, four elementary quantum gates are used [1]: $N$ gate, $F$ gate, Controlled-$V$ and Controlled-$V^+$ gates, where $VV = V^+ V^+ = N$, $VV^+ = V^+ V = I$, where $I$ is the identity. The cost of a 2-qubit gate is much larger than that of a 1-qubit gate. In this paper, the cost of $N$ gate is ignored as in [30] to be able to compare results, and assume that the cost of any 2-qubit gate is equal to 1. The cost of a reversible circuit is measured by the number of 2-qubit gates used in its implementation as a quantum circuit, so, the quantum cost of $T$ gate is 5 as shown in Fig.3

### 2.3 Basic Notions

In this section, the basic notions for reversible circuit, permutation group, NFT library, the relationship between reversible logic circuits and permutation group theory will be reviewed.
**Definition 2.2** Let \( X = \{0, 1\} \). A Boolean function \( f \) with \( n \) input variables, \( x_1, \ldots, x_n \), and \( n \) output variables, \( y_1, \ldots, y_n \), is a function \( f : X^n \to X^n \), where \( (x_1, \ldots, x_n) \in X^n \) is called the input vector and \( (y_1, \ldots, y_n) \in X^n \) is called the output vector.

**Definition 2.3** An \( n \)-input \( n \)-output Boolean function is reversible (\( n \times n \) function) if it maps each input vector to a unique output vector; i.e. a one-to-one, onto function (bijection).

There are \( 2^n! \) reversible \( n \times n \) Boolean functions. For \( n = 3 \), there are 40,320 3-input 3-output reversible functions.

**Definition 2.4** An \( n \)-input \( n \)-output reversible gate (or circuit) is a gate that realizes an \( n \times n \) reversible function.

**Definition 2.5** A set of reversible gates that can be used to build a reversible circuit is called a gate library \( L \).

**Definition 2.6** A universal reversible gate library \( L_n \) is a set of reversible gates that can be used to build any reversible circuit with \( n \)-input \( n \)-output.

**Definition 2.7** Consider a finite set \( A = \{1, 2, \ldots, N\} \) and a bijection \( \sigma : A \to A \), then \( \sigma \) can be written as,

\[
\sigma = \begin{pmatrix}
1 & 2 & 3 & \ldots & N \\
\sigma(1) & \sigma(2) & \sigma(3) & \ldots & \sigma(N)
\end{pmatrix},
\]

i.e. \( \sigma \) is a permutation of \( A \). Let \( A \) be an ordered set, then the top row can be eliminated and \( \sigma \) can be written as,

\[
(\sigma(1), \sigma(2), \sigma(3), \ldots, \sigma(N)).
\]

Any reversible circuit with \( n \)-input \( n \)-output can be considered as a permutation \( \sigma \) and Eqn.5 is called the specification of this reversible circuit such that \( N = 2^n \).

---

**Figure 4:** (a.) Circuit implementation of a reversible function. (b.) Reversible truth table.

---

The set of all permutations on \( A \) forms a symmetric group on \( A \) under composition of mappings \[10\], denoted by \( S_N \) \[4\]. A permutation group \( G \) is a subgroup \[10\] of the symmetric group \( S_N \). A universal reversible gate library \( L_n \) is called the generators of the group. Another important notation of a permutation is the product of disjoint cycles \[4\]. For example, \( (1, 2, 3, 4, 5, 6, 7, 8) \) will be written as \((1, 3, 5, 6)(7, 8)\). The identity mapping "()" is called the unit element in a permutation group. A product \( p \times q \) of two permutations \( p \) and \( q \) means applying mapping \( p \) then \( q \), which is equivalent to cascading \( p \) and \( q \).

The one-to-one correspondence between a \( n \times n \) reversible circuit and a permutation on \( A = \{1, 2, \ldots, N\} \) is established as done in \[30\]. In the permutation group references, \( A \) begins from one, instead of zero. Therefore, we have the following relation \[30\]: \( < X_N, X_{N-1}, \ldots, X_1 >_2 = \text{index}(X_N, \ldots, X_1) - 1 \). Using the integer coding, a permutation is considered as a bijective function \( f : \{1, 2, \ldots, N\} \to \{1, 2, \ldots, N\} \).
Cascading two generators is equivalent to multiplying two permutations. In what follows, a $n \times n$ reversible gate is not distinguished from a permutation in $S_N$. The set of 12 gates shown in Fig. 2 and performs as shown in Eqn.3 can be re-written as product of disjoint cycles as follows.

\[
\begin{align*}
N_1 : (1, 5)(2, 6)(3, 7)(4, 8) \\
N_2 : (1, 3)(2, 4)(5, 7)(6, 8) \\
N_3 : (1, 2)(3, 4)(5, 6)(7, 8) \\
F_{12} : (5, 7)(6, 8) \\
F_{13} : (5, 6)(7, 8) \\
F_{23} : (3, 4)(7, 8) \\
F_{21} : (3, 7)(4, 8) \\
F_{32} : (2, 4)(6, 8) \\
F_{31} : (2, 6)(4, 8) \\
T_{123} : (7, 8) \\
T_{132} : (6, 8) \\
T_{321} : (4, 8)
\end{align*}
\]

3 Schreier-Sims Algorithm

Let $A$ be a finite set. The Symmetric group, $Sym(A)$, is the group of all bijections from $A$ to itself. A permutation group $G$ is a subgroup of $Sym(A)$. Let $G \leq Sym(A)$. For $\alpha \in A$, let $G_\alpha$ denotes the stabilizer of $\alpha$ in $G$, i.e., $G_\alpha = \{ g \in G | \alpha^g = \alpha \}$. Let $B = (\alpha_1, \alpha_2, \ldots, \alpha_k)$ with $\alpha_i \in G$, then $B$ is a base for $G$ if $G(\alpha_1, \alpha_2, \ldots, \alpha_k) = 1$, and the chain of subgroups,

\[
G = G^{(1)} \geq G^{(2)} \geq \ldots \geq G^{(k+1)} = \{1\},
\]

defined by $G^{(i+1)} = G^{(i)}_{\alpha_i}$ for $1 \leq i \leq k$ is called the stabilizer chain for $B$. The orbit of $\alpha$ under $G$, denoted $\alpha^G$ is the set $\alpha^G = \{ \alpha^g | g \in G \}$. A strong generating set for $G$ relative to $B$, denoted $S$, is a set $S \subseteq G$ and for every $i$ with $1 \leq i \leq k + 1$, $G^{(i)} = \langle S \cap G^{(i)} \rangle$ holds.

This allows the problem of synthesizing a minimal number of gates (generators) to be reduced to a strong generating set-finding problem, that is, given a group $G$ acting faithfully on a finite set $A$ of size $N$. $G$ is specified by means of a generating set $S$ where each element of $S$ is expressed as a permutation on $A$.

Schreier-Sims Algorithm is a poly-time algorithm [27] known in computational Group Theory that solves the strong generating set-finding problem and also go further to solve membership testing problem. Schreier-Sims Algorithm keeps computing the generating sets using cosets, by trimming down the size of the generating set at every step using a depth first search approach to keep the size of the generating set from growing too large. An implementation of Schreier-Sims Algorithm on GAP [12] has been used to find the minimal number of generators that generates a specification of a reversible circuit given that the generators of the group can represent that specification.

Given the universal gate library NFT with 12 generators as shown in Eqn.6 and the 40320 specifications for all 3-input 3-output reversible circuits. The aim of the experiment is to answer the questions shown in the aim of the paper. To answer these questions, all sub-libraries of NFT library has been generated, that is 4095 sub-libraries after excluding the identity mapping. Use every sub-library to try to synthesize a reversible circuit for the 40320 specifications, if possible, using Schreier-Sims Algorithm. The term “if possible” here means that if a specification does not belong to the group generated by a sub-library, then it is impossible for this specification to be represented as a reversible circuit using this sub-library. The process of synthesizing all possible 3-bit reversible circuits is shown in Algorithm 1.
Algorithm 1 Generation of All 3 bits reversible circuits

1: procedure ()
2:    for \( i \leftarrow 1, \text{NumOfSubLibraries} \) do
3:        for \( j \leftarrow 1, \text{NumOfCircuits} \) do
4:            if \( \text{Specs}[j] \in G[i] \) then
5:                \( \text{Circuit}[i][j] = \text{SchreierSims}(G[i], \text{Specs}[j]) \)
6:            else
7:                \( \text{Circuit}[i][j] = [] \)
8:        end if
9:    end for
10: end for
11: end procedure

4 Experimental Results

Among the \( 4,095 \times 40,319 = 165,106,305 \) trials to synthesize a circuit for a specification using a sub-library, 80,925,627 circuits have been synthesized. The trivial specification "()" and the trivial identity mapping have been ignored. The results have been organized to answer the specification related questions (Question 1 to Question 4) in Section 4.1 and the question related to the sub-libraries (Question 5) has been answered in Section 4.2.

### 4.1 Specification Oriented Results

In this section, the results are organized to show a summary for the specification oriented results in Table 1. The results related to the bounds on circuits length are shown in Tables 2, 3, 4 and 5. Then the results related to the bounds on circuits cost are shown in Tables 6, 7, 8 and 9.

| #lib | #specs | #lib | #specs | #lib | #specs |
|------|--------|------|--------|------|--------|
| 1960 | 29070  | 2265 | 108    | 2468 | 12     |
| 1984 | 216    | 2264 | 24     | 2496 | 6      |
| 2016 | 1746   | 2266 | 80     | 2525 | 24     |
| 2044 | 36     | 2268 | 12     | 2528 | 27     |
| 2080 | 1170   | 2274 | 12     | 2540 | 6      |
| 2085 | 1      | 2284 | 48     | 2560 | 27     |
| 2086 | 559    | 2287 | 135    | 2605 | 12     |
| 2116 | 640    | 2311 | 27     | 2624 | 13     |
| 2120 | 96     | 2320 | 210    | 2625 | 8      |
| 2122 | 2974   | 2324 | 24     | 2636 | 6      |
| 2128 | 162    | 2335 | 9      | 2666 | 12     |
| 2132 | 192    | 2348 | 12     | 2688 | 9      |
| 2144 | 78     | 2353 | 24     | 2689 | 3      |
| 2152 | 36     | 2354 | 12     | 2705 | 3      |
| 2176 | 24     | 2393 | 45     | 2732 | 6      |
| 2191 | 27     | 2412 | 72     | 2816 | 7      |
| 2196 | 96     | 2417 | 27     | 2880 | 6      |
| 2198 | 12     | 2428 | 24     | 2944 | 6      |
| 2217 | 1293   | 2432 | 11     | 2961 | 3      |
| 2218 | 108    | 2450 | 6      | 3072 | 3      |
| 2220 | 72     | 2455 | 27     | 3200 | 3      |
| 2246 | 12     | 2456 | 3      | 3264 | 6      |
| 2249 | 36     | 2465 | 3      |       |        |

Table 1: There are \#specs specifications that can be synthesized by \#lib sub-libraries.

Table 1 shows the number of sub-libraries that can be used to synthesize a circuit for a specification, for example, there are 1960 sub-libraries that can be used to synthesize a circuit for 29070 specifications. There are 6 popular specifications that can be synthesized by 3264 different sub-libraries. These 6 specifications are the specifications of \( F_{12}, F_{13}, F_{23}, F_{21}, F_{32} \) and \( F_{31} \). Fig.5 shows the worst circuit
representation for $F_{31}$ with 10 gates and cost = 24. This shows that any $F$ gates can be removed from a library and be replaced by another set of generators although there might be an increase in the cost of the circuit.

![Figure 5: The worst circuit representation for $F_{31}$.](image)

| Max Len | #specs |
|---------|--------|
| 7       | 3      |
| 8       | 7      |
| 9       | 16     |
| 10      | 29     |
| 11      | 63     |
| 12      | 154    |
| 13      | 1455   |
| 14      | 14966  |
| 15      | 19544  |
| 16      | 3815   |
| 17      | 243    |
| 18      | 24     |

Table 2: There are #specs that have circuits with maximum length = Maxlen. The average maximum length = 14.639.

Table 2 shows the maximum length for the 40319 specifications, after excluding the identity. There are 3 specifications with maximum length = 7 gates and 24 specifications with maximum length = 18 gates. The average maximum length is 14.639. Table 3 shows the same results as Table 2 with more details on the cost of the maximum length, for example, there are 3 specifications with maximum length = 7 gates, the cost of two of them is 9, while the third specification has cost = 11.

Table 4 shows the minimum length for the 40319 specifications, after excluding the identity. There are 12 specifications with minimum length = 1 gate, these are the generators shown in Eqn.6 while there are 577 specifications with minimum length = 8 gates. These results are identical with that shown in [30]. This comes from the fact that all minimum length circuits come from using the gate library that contains all the 12 generators shown in Eqn.6. This is not true when finding a library of gates that synthesize minimum cost circuits as shown later. Table 5 shows the same results as Table 4 with more details on the cost of the minimum length circuit.

Table 6 shows the maximum cost for the 40319 specifications, after excluding the identity. There is 1 specification with maximum cost = 18 and 2 specifications with maximum cost = 54. The average maximum cost is 40.759. Table 7 shows the same results as Table 2 with more details on the length of the maximum cost, for example, the length of the only specification with maximum cost = 18 is 6 gates and the length of the two specifications with maximum cost = 54 are 15 gates.

Table 8 shows the minimum cost for the 40319 specifications, after excluding the identity. There are 7 specifications with minimum cost = 0, these are the cost of specifications synthesized by circuits contain the gates $N_1$, $N_2$ and $N_3$ only. There are 7 such circuits starting from a circuit with only one $N$ gate to a circuit consists of the three gates: $(N_1N_2N_3)$. There are 2049 specifications with minimum cost = 17. The average minimum cost is 11.769. This is better that that shown in [30], this enhancement comes from considering all sub-libraries. Table 9 shows the same results as Table 8 with more details on the length of the circuits with minimum cost.

Comparing the above results related to the bounds on circuits length with the results related to the bound on circuits cost shows that the cost and the length of a circuit using a library is not correlated if all types of generators are considered, not only the gates used to calculate the cost, e.g. $T$ gate is counted as 1 in circuit length while it is counted 5 in circuit cost. For example, when searching for the best length circuit for the specification $((1, 7)(2, 5)(3, 6, 8, 4))$, the best length circuit is $(F_{12}T_{32}T_{123}N_1F_{13}F_{32}T_{123})$.
| Max Len | cost | #specs | Max Len | cost | #specs | Max Len | cost | #specs | Max Len | cost | #specs |
|---------|------|--------|---------|------|--------|---------|------|--------|---------|------|--------|
| 7       | 9    | 2      | 13      | 19   | 64     | 14      | 39   | 122    | 16      | 22   | 3      |
| 7       | 11   | 1      | 13      | 20   | 37     | 14      | 40   | 107    | 16      | 23   | 18     |
| 8       | 14   | 3      | 13      | 21   | 40     | 14      | 41   | 109    | 16      | 24   | 251    |
| 8       | 20   | 1      | 13      | 22   | 187    | 14      | 42   | 62     | 16      | 25   | 102    |
| 9       | 11   | 1      | 13      | 23   | 213    | 14      | 43   | 46     | 16      | 26   | 6      |
| 9       | 13   | 2      | 13      | 24   | 72     | 14      | 44   | 52     | 16      | 27   | 66     |
| 9       | 17   | 3      | 13      | 25   | 13     | 14      | 45   | 48     | 16      | 28   | 335    |
| 9       | 19   | 6      | 13      | 26   | 59     | 14      | 46   | 9      | 16      | 29   | 505    |
| 9       | 21   | 1      | 13      | 27   | 148    | 14      | 47   | 1      | 16      | 30   | 189    |
| 9       | 24   | 3      | 13      | 28   | 124    | 14      | 48   | 16     | 16      | 31   | 15     |
| 10      | 16   | 3      | 13      | 29   | 29     | 14      | 49   | 6      | 16      | 32   | 141    |
| 10      | 18   | 1      | 13      | 30   | 43     | 15      | 19   | 24     | 16      | 33   | 728    |
| 10      | 22   | 9      | 13      | 31   | 74     | 15      | 20   | 29     | 16      | 34   | 836    |
| 10      | 24   | 9      | 13      | 32   | 82     | 15      | 21   | 26     | 16      | 35   | 167    |
| 10      | 26   | 7      | 13      | 33   | 11     | 15      | 22   | 102    | 16      | 36   | 9      |
| 11      | 11   | 2      | 13      | 34   | 7      | 15      | 23   | 1126   | 16      | 37   | 16     |
| 11      | 17   | 6      | 13      | 35   | 19     | 15      | 24   | 1392   | 16      | 38   | 128    |
| 11      | 19   | 1      | 13      | 36   | 50     | 15      | 25   | 473    | 16      | 39   | 97     |
| 11      | 21   | 12     | 13      | 37   | 26     | 15      | 26   | 76     | 16      | 40   | 17     |
| 11      | 22   | 2      | 13      | 38   | 19     | 15      | 27   | 1364   | 16      | 41   | 34     |
| 11      | 23   | 14     | 13      | 39   | 20     | 15      | 28   | 3963   | 16      | 42   | 17     |
| 11      | 24   | 2      | 13      | 40   | 2      | 15      | 29   | 2517   | 16      | 43   | 16     |
| 11      | 26   | 6      | 13      | 41   | 4      | 15      | 30   | 376    | 16      | 44   | 31     |
| 11      | 27   | 18     | 13      | 43   | 2      | 15      | 31   | 277    | 16      | 45   | 22     |
| 12      | 14   | 6      | 14      | 16   | 7      | 15      | 32   | 1663   | 16      | 46   | 14     |
| 12      | 16   | 8      | 14      | 18   | 110    | 15      | 33   | 2718   | 16      | 47   | 17     |
| 12      | 17   | 2      | 14      | 19   | 132    | 15      | 34   | 1047   | 16      | 48   | 6      |
| 12      | 18   | 21     | 14      | 20   | 158    | 15      | 35   | 60     | 16      | 49   | 21     |
| 12      | 21   | 14     | 14      | 21   | 109    | 15      | 36   | 115    | 16      | 50   | 2      |
| 12      | 22   | 27     | 14      | 22   | 662    | 15      | 37   | 317    | 16      | 51   | 1      |
| 12      | 23   | 1      | 14      | 23   | 1735   | 15      | 38   | 361    | 16      | 53   | 2      |
| 12      | 24   | 7      | 14      | 24   | 1275   | 15      | 39   | 137    | 16      | 55   | 12     |
| 12      | 26   | 17     | 14      | 25   | 254    | 15      | 40   | 174    | 16      | 57   | 18     |
| 12      | 27   | 8      | 14      | 26   | 404    | 15      | 41   | 222    | 16      | 59   | 3      |
| 12      | 28   | 4      | 14      | 27   | 2081   | 15      | 42   | 219    | 16      | 62   | 9      |
| 12      | 29   | 8      | 14      | 28   | 3115   | 15      | 43   | 195    | 16      | 63   | 12     |
| 12      | 30   | 2      | 14      | 29   | 976    | 15      | 44   | 238    | 16      | 64   | 63     |
| 12      | 31   | 6      | 14      | 30   | 129    | 15      | 45   | 98     | 16      | 66   | 35     |
| 12      | 32   | 10     | 14      | 31   | 493    | 15      | 46   | 63     | 16      | 66   | 6      |
| 12      | 33   | 4      | 14      | 32   | 1389   | 15      | 47   | 32     | 16      | 69   | 45     |
| 12      | 34   | 6      | 14      | 33   | 702    | 15      | 48   | 47     | 16      | 70   | 9      |
| 12      | 38   | 3      | 14      | 34   | 46     | 15      | 49   | 57     | 16      | 72   | 3      |
| 13      | 15   | 5      | 14      | 35   | 86     | 15      | 50   | 22     | 16      | 78   | 6      |
| 13      | 16   | 2      | 14      | 36   | 137    | 15      | 51   | 6      | 16      | 83   | 6      |
| 13      | 17   | 49     | 14      | 37   | 263    | 15      | 53   | 18     | 16      | 84   | 9      |
| 13      | 18   | 52     | 14      | 38   | 126    | 16      | 20   | 3      | 16      | 90   | 9      |

Table 3: There are #specs specifications with maximum length= MaxLen and cost = cost.
Table 4: There are \#specs specifications with minimum length = MinLen.

| Min Len |  #specs | #specs |
|---------|---------|--------|
| 1       | 12      | 12     |
| 2       | 102     | 102    |
| 3       | 625     | 625    |
| 4       | 2780    | 2780   |
| 5       | 8921    | 8921   |
| 6       | 17049   | 17049  |
| 7       | 10253   | 10253  |
| 8       | 577     | 577    |
| Avg.    | 5.865   | 5.865  |

Table 5: There are \#specs specifications with minimum length= MinLen and cost = cost.

| Min Len | cost | #specs |
|---------|------|--------|
| 1       | 0    | 3      |
| 1       | 1    | 6      |
| 1       | 5    | 3      |
| 2       | 0    | 3      |
| 2       | 1    | 24     |
| 2       | 2    | 24     |
| 2       | 5    | 15     |
| 2       | 6    | 30     |
| 2       | 9    | 3      |
| 2       | 10   | 3      |
| 3       | 0    | 1      |
| 3       | 1    | 18     |
| 3       | 2    | 117    |
| 3       | 3    | 31     |
| 3       | 5    | 24     |
| 3       | 6    | 10     |
| 3       | 6    | 162    |
| 3       | 7    | 138    |
| 3       | 9    | 18     |
| 3       | 10   | 51     |
| 3       | 11   | 39     |
| 3       | 14   | 5      |
| 3       | 15   | 1      |
| 3       | 2    | 51     |
| 3       | 3    | 282    |
| 3       | 4    | 60     |
| 3       | 5    | 8      |

Table 6: There are \#specs specifications with maximum cost = MaxCost. The average maximum cost is 40.759.
| Max Cost | Len | #specs | Max Cost | Len | #specs | Max Cost | Len | #specs | Max Cost | Len | #specs |
|----------|-----|--------|----------|-----|--------|----------|-----|--------|----------|-----|--------|
| 18       | 6   | 1      | 30       | 13  | 1      | 37       | 11  | 207    | 44       | 14  | 1652   |
| 19       | 6   | 1      | 31       | 7   | 1      | 37       | 12  | 575    | 44       | 13  | 946    |
| 20       | 6   | 1      | 31       | 9   | 1      | 37       | 13  | 1268   | 44       | 15  | 494    |
| 22       | 7   | 1      | 31       | 10  | 25     | 37       | 14  | 864    | 44       | 16  | 28     |
| 22       | 10  | 3      | 31       | 11  | 29     | 37       | 15  | 127    | 44       | 12  | 351    |
| 22       | 11  | 3      | 31       | 12  | 33     | 37       | 16  | 3     | 44       | 11  | 13     |
| 22       | 12  | 3      | 31       | 13  | 27     | 38       | 10  | 72     | 45       | 11  | 1      |
| 23       | 7   | 2      | 31       | 14  | 5      | 38       | 11  | 376    | 45       | 12  | 98     |
| 23       | 8   | 4      | 31       | 15  | 1      | 38       | 12  | 745    | 45       | 13  | 215    |
| 23       | 11  | 2      | 32       | 8   | 1      | 38       | 13  | 972    | 45       | 14  | 921    |
| 23       | 12  | 1      | 32       | 10  | 36     | 38       | 14  | 642    | 45       | 15  | 394    |
| 24       | 7   | 4      | 32       | 11  | 33     | 38       | 15  | 181    | 45       | 16  | 23     |
| 24       | 8   | 7      | 32       | 12  | 98     | 38       | 16  | 47     | 46       | 12  | 31     |
| 24       | 9   | 1      | 32       | 13  | 73     | 39       | 10  | 24     | 46       | 13  | 109    |
| 24       | 10  | 3      | 32       | 14  | 44     | 39       | 11  | 305    | 46       | 14  | 405    |
| 24       | 11  | 2      | 32       | 15  | 9      | 39       | 12  | 983    | 46       | 15  | 323    |
| 24       | 12  | 1      | 33       | 8   | 2      | 39       | 13  | 1467   | 46       | 16  | 45     |
| 25       | 7   | 18     | 33       | 9   | 15     | 39       | 14  | 830    | 47       | 12  | 22     |
| 25       | 8   | 2      | 33       | 10  | 18     | 39       | 15  | 132    | 47       | 13  | 168    |
| 25       | 9   | 2      | 33       | 11  | 54     | 39       | 16  | 28     | 47       | 14  | 275    |
| 26       | 7   | 5      | 33       | 12  | 193    | 40       | 10  | 2      | 47       | 15  | 204    |
| 26       | 8   | 2      | 33       | 13  | 173    | 40       | 11  | 83     | 47       | 16  | 18     |
| 26       | 9   | 1      | 33       | 14  | 51     | 40       | 12  | 631    | 48       | 13  | 147    |
| 26       | 10  | 10     | 33       | 15  | 43     | 40       | 13  | 1682   | 48       | 14  | 529    |
| 27       | 7   | 14     | 34       | 8   | 1      | 40       | 14  | 1437   | 48       | 15  | 230    |
| 27       | 8   | 2      | 34       | 9   | 1      | 40       | 15  | 261    | 48       | 16  | 8      |
| 27       | 9   | 5      | 34       | 10  | 16     | 40       | 16  | 15     | 49       | 13  | 1      |
| 27       | 11  | 18     | 34       | 11  | 88     | 41       | 11  | 58     | 49       | 14  | 296    |
| 27       | 12  | 2      | 34       | 12  | 186    | 41       | 12  | 378    | 49       | 15  | 303    |
| 28       | 8   | 8      | 34       | 13  | 168    | 41       | 13  | 1297   | 49       | 16  | 22     |
| 28       | 9   | 3      | 34       | 14  | 41     | 41       | 14  | 1685   | 50       | 13  | 1      |
| 28       | 10  | 3      | 34       | 15  | 24     | 41       | 15  | 411    | 50       | 14  | 10     |
| 28       | 11  | 6      | 35       | 9   | 2      | 41       | 16  | 25     | 50       | 15  | 83     |
| 28       | 12  | 3      | 35       | 10  | 8      | 42       | 11  | 107    | 50       | 16  | 2      |
| 29       | 7   | 2      | 35       | 11  | 127    | 42       | 12  | 854    | 51       | 14  | 9      |
| 29       | 8   | 1      | 35       | 12  | 370    | 42       | 13  | 1391   | 51       | 15  | 18     |
| 29       | 9   | 4      | 35       | 13  | 406    | 42       | 14  | 1305   | 51       | 16  | 1      |
| 29       | 10  | 7      | 35       | 14  | 39     | 42       | 15  | 395    | 52       | 14  | 1      |
| 29       | 11  | 6      | 36       | 10  | 2      | 42       | 16  | 17     | 52       | 15  | 4      |
| 29       | 12  | 17     | 36       | 11  | 98     | 43       | 11  | 87     | 53       | 14  | 22     |
| 30       | 8   | 2      | 36       | 12  | 509    | 43       | 12  | 1251   | 53       | 15  | 44     |
| 30       | 9   | 3      | 36       | 13  | 839    | 43       | 13  | 2241   | 53       | 16  | 2      |
| 30       | 10  | 38     | 36       | 14  | 292    | 43       | 14  | 1457   | 54       | 15  | 2      |
| 30       | 11  | 7      | 36       | 15  | 29     | 43       | 15  | 355    | 54       | 16  | 15     |
| 30       | 12  | 15     | 37       | 10  | 44     | 43       | 16  | 15     | 54       | 15  | 2      |

Table 7: There are #specs specifications with maximum cost = MaxCost and length = Len.
| Min Cost | #specs | #specs |
|----------|--------|--------|
| 0        | 7      | 7      |
| 1        | 48     | 48     |
| 2        | 192    | 192    |
| 3        | 408    | 408    |
| 4        | 480    | 480    |
| 5        | 288    | 288    |
| 6        | 592    | 592    |
| 7        | 1962   | 2016   |
| 8        | 3887   | 4128   |
| 9        | 2916   | 2496   |
| 10       | 1299   | 672    |
| 11       | 3683   | 2880   |
| 12       | 7221   | 7488   |
| 13       | 6059   | 7488   |
| 14       | 1465   | 384    |
| 15       | 3562   | 1600   |
| 16       | 7221   | 7488   |
| 17       | 2049   | 3584   |
| Avg      | 11.76967 | 11.98313 |

Table 8: There are \#specs specifications with minimum cost = MinCost.

| Min Cost | Len | #specs |
|----------|-----|--------|
| 0        | 1   | 3      |
| 0        | 2   | 3      |
| 0        | 3   | 1      |
| 1        | 1   | 6      |
| 1        | 2   | 24     |
| 1        | 3   | 18     |
| 2        | 2   | 24     |
| 2        | 3   | 117    |
| 2        | 4   | 51     |
| 3        | 3   | 51     |
| 3        | 4   | 282    |
| 3        | 5   | 75     |
| 4        | 4   | 60     |
| 4        | 5   | 375    |
| 4        | 6   | 45     |
| 4        | 7   | 1994   |
| 5        | 2   | 15     |
| 5        | 3   | 30     |
| 5        | 4   | 30     |
| 5        | 5   | 39     |
| 5        | 6   | 171    |
| 6        | 2   | 24     |
| 6        | 3   | 132    |
| 6        | 4   | 222    |
| 6        | 5   | 150    |
| 6        | 6   | 50     |
| 6        | 7   | 14     |
| 7        | 3   | 99     |

Table 9: There are \#specs specifications with minimum cost = MinCost and length = Len.
with length = 7 and quantum cost = 18, while searching for the best cost circuit for the same specification, the best cost circuit is \((N_2N_3T_{321}N_2F_{13}T_{321}N_2N_3F_{12}T_{321}N_2N_3)\) with length = 12 and cost = 17.

### 4.2 Library Oriented Results

In this section, the results are organized to show a summary of library oriented results in Table 10. The results related to the bounds on circuits maximum length synthesized by the sub-libraries are shown in Tables 11, 12 and 13. Then results related to the bounds on circuits maximum cost synthesized by the sub-libraries are shown in Tables 14, 15 and 16. The results of minimum cost and minimum circuit length are trivial results, since all sub-libraries can synthesize a circuit of length 1 and cost 0, 1 or 5.

| #lib | #specs |
|------|--------|
| 1    | 4      |
| 1    | 1342   |
| 5    | 5      |
| 9    | 35     |
| 12   | 1      |
| 15   | 11     |
| 18   | 167    |
| 24   | 71     |
| 24   | 119    |
| 27   | 143    |
| 30   | 3      |
| 37   | 23     |
| 72   | 383    |
| 73   | 7      |
| 90   | 15     |
| 90   | 31     |
| 90   | 47     |
| 90   | 63     |
| 96   | 127    |
| 99   | 95     |
| 108  | 575    |
| 117  | 719    |
| 125  | 1343   |
| 162  | 5039   |
| 168  | 1439   |
| 186  | 191    |
| 360  | 1151   |
| 1960 | 46319  |

Table 10: There are #Lib sub-libraries that can synthesize a circuit for #specs specifications.

Table 10 shows the ability of a sub-library to synthesize a circuit for a specification. It can be seen that there are 1960 sub-libraries that can be used to synthesize a circuit of any specification, i.e. there are 1960 universal reversible sub-libraries from the main library shown in Eqn.6. It is shown in the previous section that the main library is the best library to synthesize a minimum length circuit, while choosing a sub-library to synthesize a minimal cost circuit is not a trivial task.

| #lib | Max Len |
|------|---------|
| 1    | 12      |
| 2    | 54      |
| 3    | 79      |
| 4    | 148     |
| 5    | 231     |
| 6    | 289     |
| 7    | 223     |
| 8    | 364     |
| 9    | 453     |
| 10   | 498     |
| 11   | 498     |
| 12   | 433     |
| 13   | 378     |
| 14   | 255     |
| 15   | 120     |
| 16   | 54      |
| 17   | 6       |
| 18   | 3       |

Table 11: There are #lib sub-libraries that can synthesize circuits with maximum length = Max Len.

Table 11 shows the maximum length circuits synthesized by a sub-library. There are 12 sub-libraries that can synthesize a circuit with maximum length = 1. These are the sub-libraries that each of them contain a single generator from the main library shown in Eqn.6 where there are 3 sub-libraries that synthesize a circuit with maximum length = 18. These three sub-libraries are \(\{N_3, F_{32}, F_{31}, T_{123}\}\), \(\{N_2, F_{23}, F_{21}, T_{132}\}\) and \(\{N_1, F_{12}, F_{13}, T_{321}\}\). These sub-libraries are three examples from the 1960 universal reversible sub-libraries. Table 12 continued in Table 13 shows the same results as Table 11 with more details on the cost of the circuits synthesized by these sub-libraries.

Table 12 shows the maximum cost circuits synthesized by a sub-library. There are 7 sub-libraries that can synthesize a circuit with maximum cost = 0. These are the sub-libraries that each of them contain a combination of \(N\) generators, where there are 2 sub-libraries that synthesize a circuit with maximum cost = 54. These two sub-libraries are \(\{N_1, N_3, F_{32}, T_{123}, T_{321}\}\) and \(\{N_1, F_{12}, T_{123}, T_{321}\}\). These are another
| Max Len | cost | #lib |
|---------|------|------|
| 2       | 3    | 10   |
| 1       | 6    | 4    |
| 5       | 3    | 6    |
| 0       | 3    | 8    |
| 18      | 1    | 2    |
| 12      | 2    | 5    |
| 38      | 2    | 6    |
| 15      | 3    | 7    |
| 11      | 4    | 8    |
| 14      | 5    | 9    |
| 27      | 6    | 10   |
| 15      | 7    | 11   |
| 39      | 8    | 12   |
| 33      | 9    | 13   |
| 44      | 10   | 14   |
| 20      | 11   | 15   |
| 33      | 12   | 16   |
| 26      | 13   | 17   |
| 42      | 14   | 18   |
| 34      | 15   | 19   |
| 27      | 16   | 20   |
| 34      | 17   | 21   |
| 27      | 18   | 22   |
| 33      | 19   | 23   |
| 33      | 20   | 24   |

Table 12: There are \#lib sub-libraries that can synthesize circuits with maximum length = MaxLen and cost = cost.
Table 13: Table 12 cont.: There are \#lib sub-libraries that can synthesize circuits with maximum length = MaxLen and cost = cost.

| Max Len | cost | #lib |
|---------|------|------|
| 12      | 21   | 52   |
| 12      | 22   | 30   |
| 12      | 23   | 17   |
| 12      | 24   | 32   |
| 12      | 25   | 44   |
| 12      | 26   | 42   |
| 12      | 27   | 14   |
| 12      | 28   | 23   |
| 12      | 29   | 41   |
| 12      | 30   | 28   |
| 12      | 31   | 6    |
| 12      | 32   | 12   |
| 12      | 33   | 13   |
| 12      | 34   | 5    |
| 12      | 35   | 9    |
| 12      | 36   | 4    |
| 12      | 37   | 4    |
| 12      | 38   | 9    |
| 13      | 16   | 1    |
| 13      | 18   | 5    |
| 13      | 19   | 63   |
| 13      | 20   | 8    |
| 13      | 21   | 15   |
| 13      | 22   | 17   |
| 13      | 23   | 36   |
| 13      | 24   | 10   |
| 13      | 25   | 35   |
| 13      | 26   | 19   |

| Max Len | cost | #lib |
|---------|------|------|
| 13      | 27   | 17   |
| 13      | 28   | 1    |
| 13      | 29   | 11   |
| 13      | 30   | 22   |
| 13      | 31   | 10   |
| 13      | 32   | 8    |
| 13      | 33   | 10   |
| 13      | 34   | 16   |
| 13      | 35   | 10   |
| 13      | 36   | 4    |
| 13      | 37   | 13   |
| 13      | 38   | 3    |
| 13      | 39   | 13   |
| 13      | 40   | 2    |
| 13      | 41   | 4    |
| 13      | 42   | 2    |
| 13      | 43   | 3    |
| 13      | 44   | 4    |
| 13      | 45   | 1    |
| 13      | 46   | 2    |
| 13      | 47   | 3    |
| 13      | 48   | 4    |
| 13      | 49   | 3    |
| 13      | 5    | 9    |
| 13      | 19   | 6    |
| 13      | 18   | 5    |
| 13      | 16   | 1    |
| 13      | 20   | 8    |
| 13      | 21   | 15   |
| 13      | 22   | 17   |
| 13      | 23   | 36   |
| 13      | 24   | 10   |
| 13      | 25   | 55   |
| 13      | 26   | 19   |

Table 14: There are \#lib libraries that can synthesize circuits with maximum cost = Maxcost.
| Max cost | Len | #lib | Max cost | Len | #lib | Max cost | Len | #lib | Max cost | Len | #lib |
|----------|-----|------|----------|-----|------|----------|-----|------|----------|-----|------|
| 0        | 1   | 7    | 15       | 7   | 22   | 25       | 6   | 11   | 31       | 12  | 12   |
| 1        | 1   | 36   | 15       | 8   | 3    | 25       | 7   | 21   | 31       | 13  | 13   |
| 2        | 2   | 27   | 15       | 9   | 2    | 25       | 8   | 22   | 32       | 8   | 1    |
| 2        | 4   | 12   | 16       | 4   | 16   | 25       | 9   | 93   | 32       | 9   | 33   |
| 3        | 3   | 24   | 16       | 5   | 12   | 25       | 10  | 21   | 32       | 10  | 36   |
| 3        | 4   | 1    | 16       | 6   | 14   | 25       | 11  | 12   | 32       | 11  | 23   |
| 3        | 5   | 29   | 16       | 7   | 3    | 26       | 6   | 3    | 32       | 12  | 89   |
| 4        | 4   | 129  | 17       | 5   | 9    | 26       | 7   | 20   | 32       | 13  | 12   |
| 4        | 6   | 9    | 17       | 7   | 10   | 26       | 9   | 51   | 32       | 15  | 1    |
| 5        | 6   | 15   | 17       | 8   | 2    | 26       | 10  | 27   | 33       | 9   | 18   |
| 5        | 7   | 30   | 18       | 4   | 1    | 26       | 11  | 25   | 33       | 10  | 28   |
| 6        | 2   | 48   | 18       | 5   | 3    | 26       | 12  | 8    | 33       | 11  | 27   |
| 6        | 6   | 38   | 18       | 6   | 26   | 27       | 7   | 20   | 33       | 12  | 4    |
| 6        | 8   | 12   | 18       | 7   | 73   | 27       | 8   | 25   | 33       | 13  | 3    |
| 7        | 3   | 54   | 18       | 8   | 36   | 27       | 9   | 47   | 33       | 14  | 35   |
| 7        | 7   | 30   | 19       | 5   | 4    | 27       | 10  | 25   | 33       | 15  | 2    |
| 7        | 8   | 4    | 19       | 6   | 9    | 27       | 11  | 73   | 34       | 9   | 17   |
| 7        | 9   | 14   | 19       | 7   | 127  | 27       | 12  | 14   | 34       | 10  | 33   |
| 8        | 4   | 18   | 19       | 8   | 7    | 27       | 13  | 3    | 34       | 11  | 21   |
| 8        | 5   | 6    | 19       | 9   | 6    | 28       | 8   | 22   | 34       | 12  | 5    |
| 8        | 6   | 12   | 20       | 5   | 5    | 28       | 9   | 53   | 34       | 13  | 6    |
| 8        | 8   | 34   | 20       | 6   | 12   | 28       | 10  | 48   | 34       | 14  | 4    |
| 8        | 9   | 19   | 20       | 7   | 22   | 28       | 11  | 15   | 34       | 15  | 6    |
| 8        | 10  | 5    | 20       | 8   | 5    | 28       | 12  | 55   | 34       | 16  | 3    |
| 9        | 5   | 42   | 20       | 9   | 2    | 28       | 13  | 33   | 35       | 9   | 10   |
| 9        | 9   | 10   | 20       | 10  | 6    | 28       | 14  | 1    | 35       | 10  | 26   |
| 10       | 3   | 4    | 21       | 6   | 19   | 29       | 7   | 6    | 35       | 11  | 41   |
| 10       | 4   | 27   | 21       | 7   | 27   | 29       | 8   | 11   | 35       | 12  | 13   |
| 11       | 3   | 17   | 21       | 8   | 12   | 29       | 9   | 85   | 35       | 13  | 6    |
| 11       | 4   | 16   | 21       | 10  | 6    | 29       | 10  | 41   | 35       | 16  | 3    |
| 11       | 5   | 39   | 21       | 11  | 12   | 29       | 11  | 44   | 36       | 9   | 3    |
| 12       | 4   | 19   | 22       | 6   | 2    | 29       | 12  | 8    | 36       | 10  | 18   |
| 12       | 5   | 8    | 22       | 7   | 45   | 29       | 14  | 7    | 36       | 11  | 18   |
| 12       | 6   | 29   | 22       | 8   | 45   | 30       | 8   | 11   | 36       | 12  | 19   |
| 13       | 5   | 9    | 23       | 6   | 2    | 30       | 9   | 24   | 36       | 13  | 13   |
| 13       | 6   | 9    | 23       | 7   | 13   | 30       | 10  | 47   | 36       | 14  | 20   |
| 14       | 3   | 2    | 23       | 8   | 23   | 30       | 11  | 26   | 37       | 10  | 25   |
| 14       | 4   | 34   | 23       | 9   | 15   | 30       | 12  | 22   | 37       | 11  | 24   |
| 14       | 6   | 12   | 23       | 10  | 3    | 30       | 15  | 1    | 37       | 12  | 20   |
| 14       | 7   | 9    | 24       | 6   | 4    | 31       | 7   | 3    | 37       | 13  | 16   |
| 15       | 3   | 1    | 24       | 7   | 5    | 31       | 8   | 1    | 37       | 14  | 17   |
| 15       | 4   | 37   | 24       | 8   | 43   | 31       | 9   | 31   | 37       | 15  | 2    |
| 15       | 5   | 8    | 24       | 9   | 19   | 31       | 10  | 33   | 38       | 10  | 9    |
| 15       | 6   | 2    | 24       | 10  | 8    | 31       | 11  | 46   | 38       | 11  | 23   |

Table 15: There are $\#\text{lib}$ libraries that can synthesize circuits with maximum cost = Max\text{cost} and length = Len.
Table 16: Table 15 cont.: There are \#\text{lib} libraries that can synthesize circuits with maximum cost = \text{Maxcost} and length = \text{Len}.

| Max cost | Len | #lib |
|----------|-----|------|
| 38       | 12  | 13   |
| 38       | 13  | 4    |
| 38       | 14  | 3    |
| 38       | 15  | 14   |
| 38       | 16  | 3    |
| 39       | 10  | 1    |
| 39       | 11  | 15   |
| 39       | 12  | 27   |
| 39       | 13  | 13   |
| 40       | 10  | 6    |
| 40       | 11  | 9    |
| 40       | 12  | 14   |
| 40       | 13  | 2    |

| Max cost | Len | #lib |
|----------|-----|------|
| 40       | 14  | 3    |
| 40       | 18  | 3    |
| 41       | 11  | 2    |
| 41       | 12  | 12   |
| 41       | 13  | 13   |
| 42       | 11  | 19   |
| 42       | 12  | 18   |
| 42       | 13  | 10   |
| 43       | 13  | 9    |
| 43       | 14  | 4    |
| 44       | 11  | 1    |
| 44       | 13  | 10   |
| 44       | 14  | 4    |
| 44       | 14  | 1    |
| 45       | 15  | 2    |
| 45       | 13  | 4    |
| 45       | 14  | 3    |
| 46       | 12  | 1    |
| 46       | 14  | 2    |
| 46       | 15  | 2    |
| 47       | 14  | 3    |
| 47       | 12  | 2    |
| 47       | 13  | 3    |

| Max cost | Len | #lib |
|----------|-----|------|
| 47       | 14  | 5    |
| 48       | 13  | 1    |
| 49       | 14  | 3    |
| 50       | 13  | 1    |
| 50       | 15  | 3    |
| 51       | 14  | 2    |
| 51       | 15  | 1    |
| 52       | 15  | 1    |
| 53       | 15  | 2    |
| 54       | 15  | 2    |

Two examples from the 1960 universal reversible sub-libraries. Table 15 continued in Table 16 shows the same results as Table 14 with more details on the cost of the circuits synthesized by these sub-libraries.

The above results show that there are 1960 sub-libraries, each can be used as a universal reversible gate library:

1. The sub-library that synthesize the best maximum length circuits is the main library,

\[ \{N_1, N_2, N_3, F_{12}, F_{13}, F_{23}, F_{21}, F_{32}, F_{31}, T_{123}, T_{321}\}, \]

where the best maximum length = 8 gates with cost = 20 and maximum cost = 23 with circuit length = 7.

2. The sub-library that synthesize the best maximum cost circuit is,

\[ \{N_1, N_2, N_3, F_{12}, F_{13}, F_{23}, F_{21}, F_{32}, F_{31}, T_{132}, T_{321}\}, \]

where the best maximum cost = 22 with circuit length = 7 and the maximum length = 9 gates with cost = 11.

3. The sub-library that synthesize the worst maximum length circuits is,

\[ \{N_3, F_{32}, F_{31}, T_{123}\}, \]

where the worst maximum length = 18 gates with cost = 36 and maximum cost = 40 with circuit length = 18.

4. The sub-library that synthesize the worst maximum cost circuit is,

\[ \{N_1, F_{12}, T_{123}, T_{321}\}, \]

where the worst maximum cost = 54 with circuit length = 15 and the maximum length = 16 gates with cost = 49.

None of the 1960 universal reversible gate libraries can synthesize a circuit with the best maximum cost which is 17 as shown in Table 8. This best maximum cost comes from sub-libraries that are not universal, for example, the sub-library,

\[ \{N_1, N_2, F_{13}, F_{23}, F_{31}, T_{123}, T_{321}\}, \]

is not universal since it can synthesize circuits for 1151 specifications only, where its best maximum cost = 17 with circuit length = 7 and the maximum length = 7 gates with cost = 8.
5 Conclusion

By reducing the representation of the reversible circuit synthesis problem to permutation group, Schreier-Sims Algorithm for the strong generating set-finding problem is used to put tight bounds on the synthesis of 3-bit reversible circuits using the NFT library. Using group-theory algebraic software GAP shows that,

1. The minimum length of a circuit ranges from 1 to 8 gate(s) with average length = 5.865.
2. The maximum length of a circuit ranges from 7 to 18 gates with average length = 14.639.
3. The minimum cost of a reversible circuit ranges from 1 to 17 with average cost = 11.769.
4. The maximum cost of a reversible circuit ranges from 18 to 54 with average cost = 40.759.

The analysis shows that there 1960 universal reversible sub-libraries from the NFT library. The upper and lower bounds on the length of the circuits come from using the universal reversible sub-libraries while the upper bound on the cost of the circuits comes from using the universal reversible sub-libraries, while the lower bounds on the cost of the circuits comes from other sub-libraries which are not necessary universal.

The same sort of analysis is applied to other libraries such as NFP, NFFr, NFPT, NFTFr and NFPFr. The tight bounds for these libraries are under preparation.

References

[1] A. Barenco, C. H. Bennett, R. Cleve, D. P. DiVincenzo, N. Margolus, P. Shor, T. Sleator, J. A. Smolin, and H. Weinfurter. Elementary gates for quantum computation. Phys. Rev. A, 52(5):3457–3467, Nov 1995.
[2] C. Bennett. Logical reversibility of computation. IBM Journal of Research and Development, 17(6):525–532, 1973.
[3] T. Beth and M. Roetteler. Quantum algorithms: Applicable algebra and quantum physics. In Quantum Information, pages 96–150. Springer, 2001.
[4] J.D. Dixon and B. Mortimer Permutation groups. New York: Springer, 1996.
[5] A. De Vos, B. Desoete, A. Adamski, P. Pietrzak, M. Sibinski, and T. Widerski. Design of reversible logic circuits by means of control gates. In Proceedings of the 10th International Workshop on Integrated Circuit Design, Power and Timing Modeling, Optimization and Simulation, pages 255-264, 2000.
[6] A. De Vos, B. Raa and L. Storme Generating the group of reversible logic gates. Journal of Physics A: Mathematical and General, 35(33): 7063–7078, 2002.
[7] A. De Vos and Y. V. Rentergem From Group Theory to Reversible Computers. International Journal of Unconventional Computing, 4(1): 79-88, 2008.
[8] A. De Vos and S. De Baerdemacker Symmetry Groups for the Decomposition of Reversible Computers, Quantum Computers, and Computers in between. Symmetry, 3(2): 305-324, 2011.
[9] A. De Vos, B. Desoete, F. Janiak, and A. Nogawski. Control gates as building blocks for reversible computers. In Proceedings of the 11th International Workshop on Power and Timing Modeling, Optimization and Simulation, pages 9201–9210, 2001.
[10] G. W. Dueck and D. Maslov. Reversible function synthesis with minimum garbage outputs. In Proceedings of the 6th International Symposium on Representations and Methodology of Future Computing Technologies, pages 154–161, 2003.
[11] E. Fredkin and T. Toffoli. Conservative logic. International Journal of Theoretical Physics, 21:219–253, 1982.
[12] The GAP Group. GAP – Groups, Algorithms, and Programming, Version 4.6.3; 2013. (http://www.gap-system.org)
[13] O. Golubitsky and S.M. Falconer and D. Maslov. Synthesis of the optimal 4-bit reversible circuits. In Proceedings of the 47th Design Automation Conference, pages 653-656, 2010.

[14] J. Gruska. Quantum Computing. McGraw-Hill, London, 1999.

[15] K. Iwama, Y. Kambayashi, and S. Yamashita. Transformation rules for designing CNOT–based quantum circuits. In Proceedings of the 39th Conference on Design Automation, pages 419–424. ACM Press, 2002.

[16] M.I. Kargapolov and Ju.I. Merzljakov Fundamentals of the theory of groups. Berlin: Springer, 1979.

[17] D. Maslov, G. W. Dueck, and D. M. Miller. Fredkin/Toffoli templates for reversible logic synthesis. In Proceedings of the ACM/IEEE International Conference on Computer-Aided Design, page 256, 2003.

[18] D. Maslov, G. W. Dueck, and D. M. Miller. Simplification of Toffoli networks via templates. In Proceedings of the 16th Symposium on Integrated Circuits and Systems Design, page 53, 2003.

[19] D. Maslov and C. Young and D. M. Miller and G. W. Dueck. Quantum circuit simplification using templates. Design, Automation and Test in Europe, pages 1208-1213, 2005.

[20] D. Maslov. Reversible logic synthesis benchmarks. [Online]. Available: http://www.cs.uvic.ca/~dmaslov/.

[21] D. M. Miller and G. W. Dueck. Spectral techniques for reversible logic synthesis. In Proceedings of the 6th International Symposium on Representations and Methodology of Future Computing Technologies, pages 56–62, 2003.

[22] D. M. Miller, D. Maslov, and G. W. Dueck. A transformation based algorithm for reversible logic synthesis. In Proceedings of the 40th Conference on Design Automation, pages 318–323, 2003.

[23] M. Nielsen and I. Chuang. Quantum Computation and Quantum Information. Cambridge University Press, Cambridge, United Kingdom, 2000.

[24] K. N. Patel, I. L. Markov, and J. P. Hayes. Efficient synthesis of linear reversible circuits. arXiv e-Print quant-ph/0302002, 2003.

[25] A. K. Prasad and V. V. Shende and K. N. Patel and I. L. Markov and J. P. Hayes. Data structures and algorithms for simplifying reversible circuits. J. Emerg. Technol. Comput. Syst., 2(4), October 2006.

[26] D. Simon. On the power of quantum computation. In Proceedings of the 35th Annual Symposium on Foundations of Computer Science, pages 116–123, 1994.

[27] A. Seress Permutation Group Algorithms. Cambridge University Press, 2002.

[28] L. Storme, A. De Vos, G. Jacobs Group Theoretical Aspects of Reversible Logic Gates. Journal of Universal Computer Science, 5(5): 307–321, 1999.

[29] T. Toffoli. Reversible computing. In W. de Bakker and J. van Leeuwen, editors, Automata, Languages and Programming, page 632. Springer, New York, 1980. Technical Memo MIT/LCS/TM-151, MIT Lab for Computer Science (unpublished).

[30] G. Yang, X. Song, W. N.N. Hung, M. A. Perkowski, and C.-J. Seo, Synthesis of reversible circuits with minimal costs. CALCOLO, 45:193–206, 2008.

[31] A. Younes and J. Miller. Representation of Boolean quantum circuits as Reed-Muller expansions. International Journal of Electronics, 91(7):431–444, 2004.