# Fast Synthesis of Exact Minimal Reversible Circuits using Group Theory

*Guowu Yang, Xiaoyu Song, and Marek A. Perkowski*

*Dept. Electrical & Computer Engineering, Portland State University, Oregon, USA.*

## Abstract

We present fast algorithms to synthesize exact minimal reversible circuits for various types of gates and costs. By reducing reversible logic synthesis problems to group theory problems, we use the powerful algebraic software GAP to solve such problems. Our approach is not only able to minimize for arbitrary cost functions of gates, but also faster than the existing approaches to reversible logic synthesis. In addition, we show that the Peres gate is a better choice than the standard Toffoli gate in libraries of universal reversible gates.

**Index Terms**: Reversible Logic, Quantum Circuits, Minimization, Group Theory, Algorithm.

## 1. Introduction

There has been recently much research effort on developing algorithms for synthesis of reversible circuits [1-6]. The previous approaches are either not optimal, time consuming or cannot be applied to 4 qubit circuits. It has been known that any 3-bit reversible gate can be synthesized using the CNT gate library [1]. In [14], an optimal approach was proposed for synthesizing 3-bit reversible gates with an average of 5.63 gates.

Group theory has been demonstrated as a powerful tool for analysis in many applications. Few preliminary works on using group theory for reversible logic synthesis have been proposed [3, 8]. Recently, some important results have been presented in [14]. GAP [11] is a mathematical analysis package for group theory applications. It is composed of a set of efficient and fast algorithms for manipulating set and group operations. It was used to prove the universality of a given reversible logic sets [8, 13].

In this paper, we describe fast GAP-based algorithms to synthesize exact minimal reversible circuits for various types of gates with various gate costs. By reducing the reversible logic synthesis problems to the group theory problems, we are able to use the power of algebraic software GAP which is especially efficient for solving group theory problems. We are able to minimize reversible circuits for arbitrary gate cost circuits. Our algorithms are orders of magnitude faster than the existing approaches to reversible logic synthesis. In addition, we show that the Peres gate is a better choice than the standard Toffoli gate in the libraries of universal reversible gates. As an important feature, our approach can handle gates in libraries with arbitrary gate-dependent costs.

## 2. Background

**Definition 1:** Let $B = \{0, 1\}$. A Boolean logic function $f$ with $w$ input variables, $B_1, \ldots, B_w$, and $w$ output variables, $P_1, \ldots, P_w$, is a function $f: B^w \rightarrow B^w$, where $(B_1, \ldots, B_w) \in B^w$ is the input vector and $(P_1, \ldots, P_w) \in B^w$ is the output vector. A Boolean logic function $f$ is *reversible* if it is a one-to-one, onto function (*bijection*). A Boolean reversible logic function with $w$ inputs and $w$ outputs is also called a *$w \times w$ reversible gate*.

Now we introduce permutation group and its relationship with reversible functions.

**Definition 2:** Let $M = \{1, 2, \ldots, n\}$. A bijection (one-to-one, and onto mapping) of M onto itself is called a permutation on $M$. The set of all permutations on $M$ forms a group [10], under composition of mappings, called a symmetric group on $M$, denoted by $S_n$ [9]. If $M$ is a set of all $2^w$ binary vectors with length $w$, the symmetric group on $M$ is denoted by $S_{2^w}$. A permutation group is just a subgroup [10] of a symmetric group.

A mapping $a$: $M \rightarrow M$ can be written as $a = \begin{pmatrix} 1, & 2, \cdots, & n \\ i_1, & i_2, \cdots, & i_n \end{pmatrix}$. We use another notation, writing it as a product of disjoint cycles [9]. For example, $\begin{pmatrix} 1,2,3,4,5,6,7,8 \\ 1,2,4,3,5,6,8,7 \end{pmatrix}$ will be written as (3, 4)(7, 8). The identity mapping "( )" (direct wiring) is called the unity element in a permutation group. As a convention, a product $a*b$ of two permutations $a$ and $b$ means applying mapping $a$ before $b$, which corresponds to cascading $a$ and $b$.

To establish a one-to-one correspondence between a $w \times w$ reversible function and a permutation on M={1, 2, …, $2^w$}, we encode a $w$-bit binary input (output) vector $< B_w, B_{w-1}, ..., B_1 >_2$ as a unique integer value $index(B_w, \cdots, B_1) = B_1 + B_2 \cdot 2^1 + B_3 \cdot 2^2 + ... + B_w \cdot 2^{w-1} + 1$. In this formula for index, we added a "1" due to the following two reasons. First, in most of the permutation group references, $M$ begins from one, instead of zero. Second, in GAP, $M$ also begins from one, instead of zero. Therefore, we have the following relation: $< B_w, B_{w-1}, ..., B_1 >_2 = index(B_w, \cdots, B_1) - 1$. Using the integer coding, we consider a permutation as a bijection function $f$: {1, 2, ..., $2^w$} $\rightarrow$ {1, 2, ... , $2^w$}. Cascading two gates is equivalent to multiplying two permutations. In what follows, we will not distinguish a $w \times w$ reversible gate from a permutation in $S_{2^w}$. If $A$ and $B$ are subsets of a symmetric group, then $A*B$ is defined as $\{a*b \mid a \in A \wedge b \in B\}$. Let $|S|$ be the size of $S$.

**Definition 3:** $w$_library is the set of $w \times w$ reversible gates which are used to synthesize $w \times w$ reversible gates, denoted as $w\_L$, or simply as $L$. We use $T(L)$ to denote a set of all $w \times w$ reversible gates that can be synthesized using gates from library $L$.

**Definition 4:** A minimum length $minl(a)$ of any element $a$ in $T(L)$ means that there exist $minl(a)$ gates in $L$ (the gates can be of the same type) such that $a$ is a cascade of these $minl(a)$ gates, and there does not exist $k$ gates in $L$ such that $k < minl(a)$ and $a$ is a cascading of these $k$ gates. A minimum length of $T(L)$ refers to the maximum value of all minimum lengths in $T(L)$, denoted as $maxl(T(L))$ or simply as $maxl(T)$. We define $T\_k = \{a \mid a \in T(L) \wedge minl(a) = k\}$, which is the set of all elements in $T(L)$ with a minimum length $k$.

**DEFINITION 5:** A synthesis of a reversible gate $g$ means that there are $n$ gates in library $L$ such that $g$ is the cascading of these $n$ gates. The cost of the circuit refers to the sum of the costs of these $n$ gates. The minimum cost $Minc(g)$ means that there exists a realization of $g$ with cost $Minc(g)$, and there does not exist a realization with cost less than $Minc(g)$. A minimum cost synthesis of $g$ is the synthesis with cost $Minc(g)$.

## 3. Algorithms

This section presents four algorithms for reversible logic synthesis. The first and second algorithms deal with minimal length problems. Some papers discussed these problems. Miller et al [7] produced some near-optimal results. Shende at al [14] gave optimal results using group theory. Our first two algorithms are similar to [14], but these algorithms were realized in GAP, and had a faster speed. The third and fourth algorithms deal with minimal cost problems, which is more reasonable in practice because the gates have different costs.

### 3.1 Minimal length Algorithms

Given a library L, Algorithm 1 will answer the following questions:

1) What reversible gates can be synthesized by $L$, i.e., what is the set $T(L)$?
2) What is the maximum length of $T(L)$?
3) How many gates have the minimum length k? And what are these gates?

Let $A(k)$ and B($k$) denote the sets of gates, and let $n(k)$ be the size of $S(k)$, $k \geq 0$. Starting from the identity gate, we perform the permutation multiplication with library. A(0)= {( )}, A(1)=A(0)$\cup$A(0)*L. For step j, we have A(j), the set of circuits with length no more than j (Lemma 1). The next step is to perform: A(j+1)= A(j)$\cup$A(j)*L, where A(j) $\subseteq$ A(j+1), until a fix-point is reached. A detailed description is given as follows.

**Algorithm *Finding_Minimum_Length* (FML):**

**Input:** Library L.

**Output:** j, n(1), … n(j), B(1), …, B(j), A(j).

1. A(0)={( )}; G=Group(L);
2. **while** n(j)≠0 **do**
3. j=j+1;
4. A(j)=A(j-1)∪A(j-1)*L;
5. B(j)=A(j) - A(j-1);
6. n(j)=|S(j)|;
7. **end while.**

**Lemma 1:** $A(k) = \{a \mid a \in T(L) \wedge minl(a) \le k\}$.

**Proof**: According to the first and fourth lines, it is obvious that $A(k)=A(0) \cup L \cup L^2 \cup \ldots \cup L^k$. Therefore, for any $a$ in A(k), $a$ is in T(L), and $minl(a) \le k$. On the other hand, if $a$ is in T(L) and $minl(a)=m \le k$ then there exist m gates $b_1, \ldots, b_m$ in L such that $a = b_1 * \ldots * b_m$.

So, $a \in L^m \subseteq A(k)$. Thus the assertion is true. □

**Theorem 1:**

(1): Algorithm FML will halt in a finite number of steps.

(2): $A(j) = T(L)$, $j = maxl(T)$, $S(k) = T\_k$, $n(k) = |\, T\_k \,|$.

**Proof**:

(1): Since $A(k) \subseteq T(L) \subseteq S_{2^w}$, these monotonically increasing sets $A(k)$ will converge to a fix-point, which means there exist $j$ such that $A(j+1) = A(j)$. So $n(j+1)=0$. Thus Algorithm FML will halt in a finite number of steps.

(2): From Lemma 1, we have $B(k) = A(k) - A(k-1) = T\_k$, $n(k) = |B(k)| = |T\_k|$. Since $n(j+1) = 0$, we have: $A(j+1) = A(j)$. Since $A(j+1) = A(j)$, we cannot add any more new gates in $A(k)$ when cascading arbitrary gates from $L$ after the gates from $A(k)$. Thus $A(j) = T(L)$, and $j = maxl(T)$. □

Given a library $L$ and an arbitrary reversible gate g, the following algorithm determines whether g can be synthesized by using gates from $L$. If yes, the algorithm will output $k$ gates

$L[c_k]$ ,…, $L[c_1]$ in $L$ such that $g = L[c_1]*\ldots*L[c_k]$, and $k = minl(g)$. Set A(0) = {( )}. $L[c_i]$ refers to the $c_i$-th element in $L$. $(L[c_i])^{-1}$ is the inverse of $L[c_i]$. In the second algorithm, sets $A(1)$, .., $A(j)$ have the same meaning in Algorithm FML.

**Algorithm *Minimum_Length_Representation* (MLR):**

**Input:** Library L, g.

**Output:** Implementation of g with minimum length k.

1. G=Group(L); flag=0; a = g;
2. **if** g in G **then**
3. flag=1;
4. compute A(k) (k=0,1,..) as FML;
5. **if** g in A(k) **then**
6. **for** i=k **downto** 1 **do**
7. find $c_i$ such that $a*(L[c_i])^{-1} \in A(i-1)$;
8. $a = a*(L[c_i])^{-1}$;
9. **endfor**;
10. **endif**;
11. **endif**;
12. **return** flag, $L[c_k]$ ,…, $L[c_1]$, k;

**Theorem 2:**

(1) In algorithm MLR, if flag=0, then gate g can not be synthesized using gates from library $L$.

(2) If algorithm MLR returns flag=1, $L[c_k]$ ,…, $L[c_1]$, then $g = L[c_1]*\ldots*L[c_k]$, $minl(g) = k$.

**Proof**: (1) If flag=0, g is not in Group(L), namely $g$ can not be generated by library $L$.

(2) From algorithm MLR, g is in $A(k)$ but not in $A(k\text{-}1)$, so $k = minl(g)$. From for loop 6 to 9, we have

$( ) = g*(L[c_k])^{-1}*\ldots*(L[c_1])^{-1}$. Therefore,

$g = [(L[c_k])^{-1}*\ldots*(L[c_1])^{-1}]^{-1} = L[c_1]*\ldots*L[c_k]$. □

### 3.2 Minimal Cost Algorithms

In practice, the costs of NOT gates, Feynman gates (called also the Controlled_NOT gates) and other well-known gates are different (for instance, minimum costs for NMR realization technology of quantum gates are given in [15]). For instance, a 3*3 reversible gate can be implemented by quantum gates: 1-qubit NOT gate, 2-qubit Feynman gate, Controlled_V gates and Controlled_$V^+$ (Hermitian) gate. The cost of a 2-qubit gate is much larger than that of a 1-qubit gate. Thus, we approximately ignore the cost of NOT gates and assume the cost of a 2-qubit gate is equal to 1. The cost of a reversible gate is measured by the number of the 2-qubit gates in its optimal implementation of quantum gates. As a result, for 3*3 reversible gates, we can have a reasonable approximation for the gate costs: cost (NOT)=0, cost(Feynman)=1, cost(Peres)=4 (see Fig.1), cost(Toffoli)=5 (see Fig.2) and cost(Fredkin)=5. Because of the different costs of the gates in a library, we cannot equally deal with each gate in the library. In the following, we present two algorithms to find and to represent the implementation with minimum cost in any given library. If in future new costs will be calculated for some quantum realization technology other than NMR [15], we can easily adapt these costs to our CAD tools for quantum synthesis.

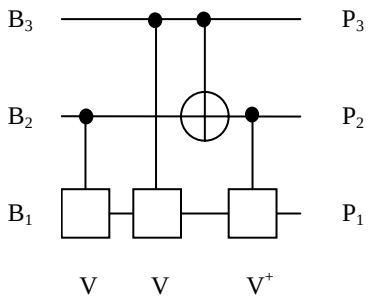


Figure 1: Peres gate $Pe_{12}$

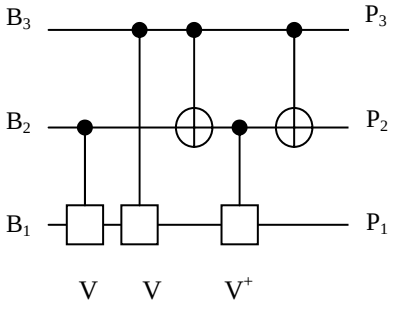


Figure 2: Toffoli gate $To_1$

Generally, assume a library $L$ includes $p$+1 parts: $L0$ = {gates with cost 0} (It always contains a direct wiring, i.e., identity gate), and $Lr_i$ = {gates with cost $r_i$}, where $r_i > 0$ are different integers, $i = 1, \ldots, p$. These two algorithms can be run with arbitrary values of costs of gates. This is practically important, since for instance it is known that costs of the same gates in quantum NMR realization, optical realization and ion trapped realization can differ considerably

from one and another and ratios of costs of the same gates vary significantly between the technologies.

The following algorithm gives the number $n(k)$ of the reversible gates with the minimum cost $k$ and the set B($k$) of these gates. Set $A(k)$ is the set of circuits with cost no more than k. A(k) and B(k) are computed as follows: A(0)=Group(L0) (All zero cost circuits),
$Lr_iN=Lr_i*A(0)$, i=1,…p,
$A(j)=A(j-1)\cup A(j-r_1)*Lr_1N\cup...\cup A(j-r_p)*Lr_pN$ (if $j-r_i<0$, then we do not need to union the product $A(j-r_i)*Lr_iN$), j=1,2,…. B(0)=A(0), B(j)=A(j)-A(j-1).

**Algorithm Finding_Minimum_Cost (FMC):**

**Input:** L0, $Lr_1$, …, $Lr_p$, $r_1$, …, $r_p$;

**Output:** j, n(0), n(1), …, n(j), B(0), B(1), …, B(j), A(j).

1. G = *Group*(L); m = |G|;
2. A(0) = *Group*(L0); B(0) = A(0);
3. **for** $1\le i\le p$ **do** $Lr_iN= \{(), Lr_i\}*B(0)$;
4. j= 1; ma = |A(0)|;
5. **while** (ma<m) **do**
6. j=j+1; A(j)=A(j-1);
7. **for** $1\le i\le p$ **do**
8. **if** $(j-r_i \ge 0)$ **then** $A(j) = A(j) \cup A(j-r_i)*Lr_iN$;
9. **endfor;**
10. ma = |A(j)|; B(j) = A(j) - A(j-1); n(j)=|B(j)|;
11. **endwhile**;

**Lemma 2:** $B(k) = \{a \mid a\in T(L), Minc(a) \le k\}$ $0\le k\le j$.

**Proof:** Case 1: For any $g\in A(k)$, the lines 2, 3, 6 and 8 tell us $g\in T(L)$, and cost of $g$ is less than or equal to $k$.

Case 2: For any $a\in T(L)$, $Minc(a) \le k$, we will prove $a\in A(k)$ by induction on $k$. When $k = 0$, the definition of $Minc(a)$ and the line 2 show that $a\in A(0)$. Suppose it is true for $k \le h$. Consider the situation $k = h+1$. If $Minc(a) \le h$, according to the assumption, it is true. If $Minc(a) = h+1$, there

exist $2n+1$ gates (i.e. the gates $b_0$, $b_1$, …, $b_{n-1}$, $b_n$ in $A(0)$ and $c_1$, …, $c_n$ in the union of $Lr_i$) such that $a = b_0*(c_1* b_1)*\ldots*(c_{n-1}*b_{n-1})*(c_n*b_n)$ and the sum of the cost of $c_i$ is $h+1$. Assuming the cost of $c_n$ be $r_i$, then $b_0*(c_1* b_1)*\ldots*(b_{n-1}*c_{n-1})$ is in $A(h+1-r_i)$. According to the line 8, we have $a \in A(h+1)$. □

## 4. Experiments

We present some experiments on 3-qubit synthesis. All experiments are running on an **850MHz Pentium® III** computer.

We first introduce some libraries and then give the experimental results. In the following, we give some permutations of the well-known 3*3 Feynman gates and NOT gates. We use ⊕ to denote XOR.

Feyman gates: $Fe_{12}$: $P_3=B_3$, $P_2=B_2$, $P_1=B_1 \oplus B_2$.

**Table 1:** Permutation of Feynman gate $Fe_{12}$

| Inputs | | | | Outputs | | | |
|---|---|---|---|---|---|---|---|
| B3 | B2 | B1 | index | P3 | P2 | P1 | index |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 2 | 0 | 0 | 1 | 2 |
| 0 | 1 | 0 | 3 | 0 | 1 | 1 | 4 |
| 0 | 1 | 1 | 4 | 0 | 1 | 0 | 3 |
| 1 | 0 | 0 | 5 | 1 | 0 | 0 | 5 |
| 1 | 0 | 1 | 6 | 1 | 0 | 1 | 6 |
| 1 | 1 | 0 | 7 | 1 | 1 | 1 | 8 |
| 1 | 1 | 1 | 8 | 1 | 1 | 0 | 7 |

$Fe_{12}$=(3,4)(7,8),

Similarly, we have $Fe_{13}$=(5,6)(7,8), $Fe_{21}$=(2,4)(6,8),

$Fe_{23}$=(5,7)(6,8), $Fe_{31}$=(2,6)(4,8), $Fe_{32}$=(3,7)(4,8).

NOT gates: $N_1$: $P_3=B_3$, $P_2=B_2$, $P_1=B_1$' (inverter of $B_1$).

$N_1$=(1,2)(3,4)(5,6)(7,8), $N_2$=(1,3)(2,4)(3,7)(4,8),

$N_3$=(1,5)(2,6)(3,7)(4,8).

3_NFT (or 3_CNT) library: this library includes 3 NOT gates, 6 Feynman gates, and 3 Toffoli gates.

$To_1$ = (7,8), i.e., $P_3=B_3$, $P_2=B_2$, $P_1=B_1 \oplus B_2B_3$,

$To_2$ = (6,8), i.e., $P_3=B_3$, $P_2=B_2 \oplus B_1B_3$, $P_1=B_1$,

$To_3$ = (4,8), i.e., $P_3=B_3 \oplus B_1B_2$, $P_2=B_2$, $P_1=B_1$.

3_NFP library: this library includes 3 NOT gates, 6 Feynman gates, and 6 Peres gates. One example of Peres gate is shown in Fig.1. In the following, we constructively demonstrate that Peres gate is a better choice in a universal library than a popularly used Toffoli gate. Not only is the gate cheaper in quantum realization, but on average circuits have a smaller number of gates and have smaller total costs when Peres gates are used instead of Toffoli gates.

According to Figure 1, we have

$Pe_{12}$=(5,7,6,8), i.e., $P_3=B_3$, $P_2=B_2 \oplus B_3$, $P_1=B_1 \oplus B_2B_3$,

Similarly, we have

$Pe_{13}$=(3,7,4,8), i.e., $P_3=B_3 \oplus B_2$, $P_2=B_2$, $P_1=B_1 \oplus B_2B_3$,

$Pe_{21}$=(5,6,7,8), i.e., $P_3=B_3$, $P_2=B_2 \oplus B_1B_3$, $P_1=B_1 \oplus B_3$,

$Pe_{31}$=(3,4,7,8), i.e., $P_3=B_3 \oplus B_1B_2$, $P_2=B_2$, $P_1=B_1 \oplus B_2$,

$Pe_{23}$=(2,6,4,8), i.e., $P_3=B_3 \oplus B_1$, $P_2=B_2 \oplus B_2B_3$, $P_1=B_1$,

$Pe_{32}$=(2,4,6,8), i.e., $P_3=B_3 \oplus B_1B_2$, $P_2=B_2 \oplus B_1$, $P_1=B_1$.

3_NFFr library: this library includes 3 NOT gates, 6 Feynman gates, and 3 Fredkin gates.

$Fr_1$ = (4,6), $P_3=B_1'B_3+B_1B_2$, $P_2=B_1'B_2+B_1B_3$, $P_1=B_1$,

$Fr_2$ = (4,7), $P_3=B_2'B_3+B_2B_1$, $P_2=B_2$, $P_1=B_2'B_1+B_2B_3$,

$Fr_3$ = (6,7), $P_3=B_3$, $P_2=B_3'B_2+B_3B_1$, $P_1=B_3'B_1+B_3B_2$.

We implemented the above algorithms using GAP. We then supplied all 40320 possible 3-bit reversible gates as specifications to be synthesized by our algorithms. Our algorithms synthesized all these 40320 gates in very short time (see Table 2, 3 and 4). Time is measured in seconds.

**Table 2:** Time of number of gates with minimal length k in different papers

| NFT Lib. | [4] 750MHz Pentium III Non-optimal solution | [7] 2GHz Pentium IV optimal solution | We: 850MHz Pentium III optimal solution |
|---|---|---|---|
| Time | 20 | 40 | 12 |

Table 3 presents results for various gate libraries: NFT, NFP, NFFr, NFPT (NOT, Feynman, Peres, Toffoli), NFTFr (NOT, Feynman, Toffoli, Fredkin) and NFPFr (NOT, Feynman, Peres, Fredkin). The parameter "aver." means the average minimum length. Observe that NFP is a winner in the category of three-gate libraries and NFPFr in the category of four-gate libraries. Cascades with Peres gates are shorter both on average and for the most complex circuits. We proved here that every 3-qubit circuit can be realized with at most 6 gates: NOT, Feynman and Peres. Our design times are better than those reported in the previous work.

**Table 3:** Number of gates with minimum length *k*

| Mini-length | NFT | NFP | NFFr | NFPT | NFTFr | NFPFr |
|---|---|---|---|---|---|---|
| 0 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 12 | 15 | 12 | 18 | 15 | 18 |
| 2 | 102 | 174 | 101 | 228 | 143 | 248 |
| 3 | 625 | 1528 | 676 | 1993 | 1006 | 2356 |
| 4 | 2780 | 8968 | 3413 | 10503 | 5021 | 12797 |
| 5 | 8921 | 23534 | 11378 | 23204 | 15083 | 22794 |
| 6 | 17049 | 6100 | 17970 | 4373 | 17261 | 2106 |
| 7 | 10253 | 0 | 6739 | 0 | 1790 | 0 |
| 8 | 577 | 0 | 30 | 0 | 0 | 0 |
| Total | 40320 | 40320 | 40320 | 40320 | 40320 | 40320 |
| Aver. | 5.87 | 4.84 | 5.66 | 4.73 | 5.33 | 4.60 |
| maxl | 8 | 6 | 8 | 6 | 7 | 6 |
| Time | 12 | 10 | 13 | 10 | 12 | 11 |

It can be also seen in Tables 3 and 4 that all libraries that include Peres gate lead to cheaper circuits than those that do not include such gate (Aver. means the average minimum cost, and we

assume that cost of NOT gate = 0, cost of Feynman gate = 1, cost of Toffoli gate = 5, cost of Peres gate = 4, and cost of Fredkin gate = 5). It can be observed that NFP library is good enough, NFPFr library is just a little better than NFP after adding 3 Fredkin gates.

**Table 4:** Number of circuits with minimum cost *k*

| Mini-cost | NFT | NFP | NFFr | NFPT | NFTFr | NFPFr |
|---|---|---|---|---|---|---|
| 0 | 8 | 8 | 8 | 8 | 8 | 8 |
| 1 | 48 | 48 | 48 | 48 | 48 | 48 |
| 2 | 192 | 192 | 192 | 192 | 192 | 192 |
| 3 | 408 | 408 | 408 | 408 | 408 | 408 |
| 4 | 480 | 672 | 480 | 672 | 480 | 672 |
| 5 | 288 | 1248 | 288 | 1248 | 384 | 1344 |
| 6 | 592 | 3184 | 880 | 3184 | 1072 | 3568 |
| 7 | 2016 | 4320 | 3008 | 4320 | 3104 | 3968 |
| 8 | 4128 | 3552 | 3904 | 3552 | 3808 | 3424 |
| 9 | 2496 | 11520 | 1440 | 11520 | 1248 | 11520 |
| 10 | 672 | 4416 | 416 | 4416 | 1856 | 4416 |
| 11 | 2880 | 0 | 4608 | 0 | 6720 | 0 |
| 12 | 7488 | 9856 | 10432 | 9856 | 7552 | 9856 |
| 13 | 7488 | 896 | 3456 | 896 | 2688 | 896 |
| 14 | 384 | 0 | 0 | 0 | 0 | 0 |
| 15 | 1600 | 0 | 0 | 0 | 6784 | 0 |
| 16 | 5568 | 0 | 4608 | 0 | 3840 | 0 |
| 17 | 3584 | 0 | 6144 | 0 | 128 | 0 |
| Aver. | 11.98 | 9.08 | 11.87 | 9.08 | 11.38 | 9.06 |
| Time | 112 | 111 | 123 | 126 | 159 | 126 |

From the library NFP and NFPT in Table 4, we know that any 3*3 reversible circuits can be realized by no more than 13 2-qubit control and XOR quantum gates and some 1-qubit NOT gates. Therefore, the maximum cost of any 3*3 reversible circuits is no more than 13.

## 5. Conclusion

By reducing the reversible circuit synthesis problem to group theory representation and using group-theory algebraic software GAP, we were able to synthesize exact 3-qubit circuits with the minimum numbers of gates from various libraries. Our approach synthesizes minimum-cost circuits from libraries of gates with arbitrary costs. We showed that a Peres gate is better than the Toffoli gate that is used by practically every research paper. We demonstrated on several examples and in an exhaustive analysis the importance and usefulness of the Peres gate. It is the cheapest gate in NMR quantum realization [15]. Using the library containing the Peres gate, we can have circuits with a smaller number of gates with a less cost. As a result, there are no more reasons to use Toffoli gates in practical NMR designs.

## 6. References


[1] T. Toffoli, *Reversible computing*, Tech. Memo MIT/LCS/TM-151, MIT Lab for Comp. Sci, 1980.

[2] V. V. Shende, A.K. Prasad, I.L. Markov, and J.P. Hayes, "Reversible logic circuit synthesis," *Proceedings of ICCAD*, pp. 125-132, San Jose, California, USA, Nov 10-14, 2002.

[3] A. De Vos, B. Raa and L. Storme, "Generating the group of reversible logic gates," *Journal of Physics A: Mathematical and General*, 35, pp. 7063–7078, 2002.

[4] R. Forf, *Artificial intelligence search algorithms*, Algorithms and Theory of Computation Handbook, CRC Press, 1999.

[5] M. Perkowski, M. Lukac, M. Pivtoraiko, P. Kerntopf, M. Folgheraiter, "A hierarchical approach to computer aided design of quantum circuits," *6th International Symposium on Representations and Methodology of Future Computing Technology*, pp. 201-209, Trier, Germany, March 2003.

[6] S. S. Bullock and I. L. Markov, "An arbitrary two-qubit computation in 23 elementary gates," *Proceedings of DAC*, pp. 324-329 Anaheim, Cal, USA, June 2003.

[7] D. M. Miller, D. Maslov and G. W. Dueck, "A transformation based algorithm for reversible logic synthesis," *Proceedings of DAC*, pp. 318-323, Anaheim, Cal, USA, June 2003.

[8] L. Storme et al, “Group theoretical aspects of reversible logic gates,” *Journal of Universal Computer Science*, 5 (1999), pp. 307-321.

[9] J. D. Dixon, and B. Mortimer, *Permutation Groups*, Springer, New York, 1996.

[10] M. I. Kargapolov, and Ju. I. Merzljakov, *Fundamentals of the Theory of Groups*, Springer-Verlag, New York, 1979.

[11] M. Schonert et.al, *GAP-Group, Algorithms, and Programming*, Lehrstuhl D fur Mathematik, Rheinisch Westfalische Technische Hochschule, Aachen, Germany, fifth, 1995.

[12] J. A. Smolin, and D. P. DiVincenzo, “Five two-bit quantum gates are sufficient to implement the quantum Fredkin gate,” *Physical Review A*, 53 (1996), pp. 2855-2856.

[13] G. Yang, W. N. N. Hung, X. Song, and M. Perkowski. “Majority-based reversible logic gate”, *6th International Symposium on Representations and Methodology of Future Computing Technology*, pp. 191-200, Trier, Germany, March 2003.

[14] V.V. Shende, A.K. Prasad, I.L.Markov, and J.P. Hayes, “Synthesis of Reversible Logic Circuits”, *IEEE Trans. on Computer Aided Design of Integrated Circuits and Systems*, Vol. 22, No. 6, June 2003, pp. 710-723.

[15] S. Lee, J.-S. Lee, S.-J. Lee, and T. Kim., “Costs of basic gates in quantum computation”, *submitted to Journal of Physics A*, (cited with author’s permission from private communication), 2004.