Bi-direction Synthesis for Reversible Circuits

Guowu Yang, Xiaoyu Song, William N. N. Hung, and Marek A. Perkowski

Dept. ECE, Portland State University, Oregon, USA.

ABSTRACT

Quantum computing is one of the most promising emerging technologies of the future. Reversible circuits are an important class of Quantum circuits. In this paper, we investigate the problem of optimally synthesizing four-qubit reversible circuits. We present an enhanced bi-directional synthesis approach. Due to the super-exponential increase on the memory requirement, all the existing methods can only perform four steps for the CNP (Control-Not gate, NOT gate, and Peres gate) library. Our novel method can achieve 12 steps. As a result, we augment the number of circuits that can be optimally synthesized by over $5*10^6$ times. Moreover, our approach is faster than the existing approaches by orders of magnitude. The promising experimental results demonstrate the effectiveness of our approach.

Categories and Subject Descriptors

M1.8 [Design Methodologies]: Logic Design

General Terms

Design, Theory

Keywords

Reversible Logic, Quantum Circuits, Minimization, Algorithm

1. INTRODUCTION

Quantum computing is one of the most promising emerging technologies in future. Reversible circuits are an important class of Quantum circuits. There has been recently some research effort on synthesis of reversible circuits [1-6]. Exact minimal results were given only for 3-qubit circuits. There are no exact results on 4-qubit circuits and all published results for more than 3-qubit circuits are only heuristics with no evaluation of distance from the minimal solution. Moreover, even these heuristic results are usually slow since they depend on algorithms such as evolutionary or simulated annealing. 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. In [2], a synthesis method of optimal circuits was proposed.

Group theory has been demonstrated as a powerful tool for analysis in many applications. 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 some given sets of reversible gates [8, 13]. Recently, more and more works on using group theory for reversible logic synthesis are being proposed [2, 3, 8, 13, 14, 15].

In this paper, we investigate the problem of optimally synthesizing four-qubit reversible circuits. We present an enhanced bi-directional synthesis approach. Due to the super-exponential increase on the memory requirement, all the existing methods can only perform four steps for the CNP library [15]. Our novel method outperforms them being able to execute 12 steps. It is also orders of magnitude faster than the existing approaches. The number of circuits that can be optimally synthesized has increased over $5*10^6$ times. The promising experimental results prove the effectiveness of our approach.

2. BACKGROUND

Definition 1: Let B = {0, 1}. A Boolean logic function f with w input variables, B_1, B_2, …, B_w, and w output variables, P_1, P_2, …, P_w, is a function $f: B^w \rightarrow B^w$, where $<B_1, B_2, ..., B_w> \in B^w$ is the input vector and $< P_1, P_2, ..., P_w > \in B^w$ 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,…, 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

Proceedings of the IEEE Computer Society Annual Symposium on VLSI
New Frontiers in VLSI Design
0-7695-2365-X/05 $20.00 © 2005 IEEE

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* → *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 ( ) is called the unity element in a permutation group. As convention, a product a*b of two permutations *a* and *b* means applying mapping *a* before *b*.

To establish a one-to-one correspondence between a reversible function and a permutation, we encode a *w*-bit binary input (output) vector $< B_w, B_{w-1}, ..., B_1 >_2$ as a unique integer value $index(B_w, \cdots, B_1) = B_1 + B_2 \cdot 2^1 + B_3 \cdot 2^2 + ... + B_w \cdot 2^{w-1} + 1$. We add one due to the following two reasons. First, in most of the permutation group books, *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} → {1, 2, ... , 2^w}. Cascading two gates is equivalent to multiplying two permutations. In what follows, we will not distinguish a *w*×*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* | *a*∈*A*∧*b*∈ *B*}. Let |*S*| be the size of *S*.

Definition 3: *w*_library is the set of *w*×*w* reversible gates which are used to synthesize *w*×*w* reversible gates, denoted as *w_L*, or simply as *L*. What a *w*×*w* reversible circuit g can be synthesized by L means that there are some gates in L such that g is the product of these gates. We use *T(L)* to denote a set of all *w*×*w* reversible circuits that can be synthesized using gates from library *L*.

Definition 4: A minimum length *l*(*a*) of any element *a* in *T*(*L*) means that there exist *l*(*a*) gates in *L* (the gates can be of the same type) such that *a* is a cascade of these *l*(*a*) gates, and there does not exist *k* gates in *L* such that *k* < *l*(*a*) and *a* is a cascading of these *k* gates.

3. ALGORITHM

In this section, we will answer these questions: Given a library *L* and an arbitrary reversible gate g, can g be synthesized by using gates from *L*? If yes, how to synthesize g with minimal length in limited memory space?

When we use breadth-first search algorithms [15] to deal with 4-qubit reversible problems, the memory becomes quickly exhausted. In our computer, the breadth-first search algorithm can only go 4 steps (the depth of search with corresponds to the length of the reversible cascade) when we use library CNP (Control-Not gate, NOT gate, and Peres gate), and only about 1 million 4-qubit reversible functions can be realized (see Table 5 in section 4). The breadth-first search method is a forward search from identity until the specification, the given reversible circuit g is found. Now we propose a bi-directional search method: a forward search from identity and a backward search from the specification g. The idea is to compute the inverse of the library L, L^{-1} = {a|a^{-1}∈L}. Set s is the maximum number of steps which a computer can go when using breadth-first search method in the computer. Compute the forward sets Af(1) = L, …, Af(s) = Af(s-1)*L, and the backward sets Ab(1) = L^{-1}*g, …, Ab(s) = L^{-1}*Ab(s-1). If Af(j)∩Ab(j-1) = ϕ, but Af(j)∩Ab(j) ≠ ϕ, then the minimum number of steps of g is 2j. If Af(j)∩Ab(j) = ϕ, but Af(j+1)∩Ab(j) ≠ ϕ, then the minimum number of steps of g is 2j+1. In the bi-directional search approach, the needed memory is only doubled, but the steps are doubled as well, up to 2*s. The number of the reversible circuits which can be constructed by library L is dramatically increased with the number of steps. The bi-directional algorithm is also significantly faster, because the number of the searched reversible circuits is much less than the forward search only (See table 4). For instance, when we synthesize a circuit with minimum length 4, we need to remember |Af(4)| = 1115774 permutations if we use forward search. But if we bi-direction search, we only need to remember |Af(2)|+|Ab(2)| = 2734 permutations. We use a simple strategy to enhance the bi-directional method such that the length or cost grows up to 2*s+t (t ≤ s). In a forward search one begins from an identity. In our enhanced bi-directional search method, we form a loop: in the forward search, we search first from the identity, then from all elements in B(1), then B(2), …, then B(t). The backward search remains unchanged. The statement in the loop is the bi-directional search. When g has not been represented, we do not need to keep the data, so the memory is not increased.

```
Algorithm: Bi-direction Search (BDS)
Input: L, g;
Output: leng, $h_1$, $h_2$, …, $h_{leng}$
1. Af(0)={()}; Ab(0)={g}; Flag=0; i=1;
2. While Flag=0 do
3.     Af(i)=Af(i-1)*L; Ab(i)=$L^{-1}$*Ab(i-1);
4.     If Af(i)∩Ab(i-1)≠ϕ then
5.         Flag=1; leng=2*i-1;
```

```
        Select a in $Af(i)\cap Ab(i-1)$, find $h_1, h_2, \dots, h_i, h_{i+1}, \dots, h_{2*i-1}$ in L such that $a=h_1*h_2*\dots*h_i=g*(h_{2*i-1})^{-1}*\dots*(h_{i+1})^{-1}$;
    Elseif $Af(i)\cap Ab(i)\neq\phi$ then
        Flag=1; leng=2*i;
        Select a in $Af(i)\cap Ab(i)$, find $h_1, h_2, \dots, h_i, h_{i+1}, \dots, h_{2*i}$ in L such that $a=h_1*h_2*\dots*h_i=g*(h_{2*i})^{-1}*\dots*(h_{i+1})^{-1}$;
    Else i=i+1;
End while.
```

Theorem 1: In the algorithm BDS, the parameter leng is the minimum length l(g), and $g=h_1*h_2*...*h_{leng}$ is the minimum length design of circuit g, where $h_j\in L$ for j=1, 2, …, leng.

Proof: According to the algorithm BDS, $g=h_1*h_2*...*h_{leng}$, where $h_j\in L$ for j=1, 2, …, leng. Thus l(g)≤leng. In the following, we will prove l(g)=leng. Then the proof of the theorem finishes.

Case 1. Suppose l(g)=2k-1, an odd number. Then there exist $b_1, b_2, \dots, b_{2k-1}\in L$ such that $g=b_1*b_2*\dots*b_k*b_{k+1}*\dots*b_{2k-1}$.

So $g*(b_{2k-1})^{-1}*\dots*(b_{k+1})^{-1}=b_1*b_2*\dots*b_k$. According the computation of Af(k) and Ab(k), $b_1*b_2*\dots*b_k\in Af(k)$, $g*(b_{2k-1})^{-1}*\dots*(b_{k+1})^{-1}\in Ab(k-1)$. Thus leng ≤ 2k-1=l(g). Combining with l(g)≤leng, leng=2k-1=l(g).

Case 2. l(g)=2k, an even number. Similar to case 1, leng=l(g).

4. Synthesis of Non-Reversible logic circuit or odd reversible circuit

This section shows how to realize any n-qubit non-reversible logic circuit or odd reversible circuit by using 1-qubit NOT gate, 2-qubit CNOT gate and 3-qubit Peres gate (CNP).

Lemma 1: All even n-qubit reversible circuits can be realized by CNP.

Proof: We know that all even n-qubit reversible circuits can be realized by 1-qubit NOT gates, 2-qubit CNOT gates and 3-qubit Toffoli gates (CNT) [2]. Toffoli gate can be realized by a Peres gate cascaded with a Feynman gate. Therefore, all even n-qubit reversible circuits can be realized by CNP. €

Now we deal with any logic circuit f, a non-reversible logic circuit or odd reversible circuit.

Theorem 2: Suppose that f is a non-reversible logic circuit. There are t inputs $B_1, \dots, B_t$, s outputs $P_1, \dots, P_s$ ($s \le t$). The truth table of f is M. In M there are at most r rows that are the same. Then by adding $\lceil \log_2 r\rceil$-(t-s) (if $\lceil \log_2 r\rceil$ > (t-s), else 0) inputs with constant zero, f can be realized by CNP with $\lceil \log_2 r\rceil$ garbage outputs.

Proof: Consider the case: t = s = n, r = 2. We add one qubit for input and output, and consider a new (n+1)-qubit circuit with truth table (Table 1):

Table 1: added Input and Output of function f

Input		Output	
B_{n+1}	$B_n, \dots, B_1$	P_{n+1}	$P_n, \dots, P_1$
0 ⋮ 0	0 ⋯ 0 ⋮ ⋱ ⋮ 1 ⋯ 1	C_1	N_1
1 ⋮ 1	0 ⋯ 0 ⋮ ⋱ ⋮ 1 ⋯ 1	C_2	N_2

The column vector $\begin{bmatrix} C_1 \\ C_2 \end{bmatrix}$ is a binary vector, and the number of zeros and the number of ones are the same. We set N_1=M, and construct C_1, C_2 and N_2 such that the rows of matrix $\begin{bmatrix} C_1 & N_1 \\ C_2 & N_2 \end{bmatrix}$ is an even permutation of the rows of the input matrix. Then the (n+1)-qubit circuit is an even reversible circuit. According to Lemma 1, this circuit can be realized by CNP. Therefore, by adding a zero constant qubit, the circuit f can be realized by CNP.

Similarly the other cases can be dealt with. €

5. EXPERIMENTS

In this section, we present some experiments on 4-qubit synthesis by using CNP (Control-Not gate, NOT gate, and Peres gate). All experiments are running on an **850MHz Pentium® III** computer.

We first introduce how many permutations will be in the library CNP (Control-Not gate, NOT gate, and Peres gate) when we use GAP.

Not gates N_i: $P_i=B_i$', $P_m=B_m$, if m≠i. (The subscript number i means the 1-qubit converter gate connected to wire i, see Fig.1) There are four connections, thus there are four permutations of NOT gate in our permutation library. For instance, $N_1=(1,2)(3,4)(5,6)(7,8)(9,10)(11,12)(13,14)(15,16)$.

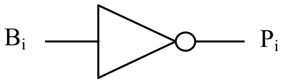

Figure 1: Not gate N_i

Control-Not gate $C_{i,j}$: $P_i=B_i\oplus B_j$, $P_m=B_m$, if $m\neq i$. (The subscript numbers i and j mean that this 2-qubit Control-Not gate is connected to the wire i and j, and the XOR is connected to the wire i, the first number of these subscript numbers, see Fig.2) There are 12 connections, thus there are 12 permutations of Control-Not gate in our permutation library. For instance, $C_{2,3}=(5,7)(6,8)(13,15)(14,16)$.

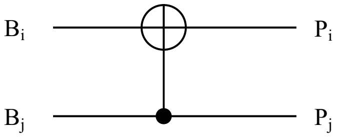

Figure 2: Control-Not gate $C_{i,j}$

Peres gate $P_{i,j,k}$: $P_i=B_i\oplus B_jB_k$, $P_j=B_j\oplus B_k$, $P_m=B_m$, if $m\neq i$, j. (see Fig.3) There are 24 connections, thus there are 24 permutations of Peres gate in our permutation library. For instance, $P_{2,3,4}=(9,13,11,15)(10,14,12,16)$.

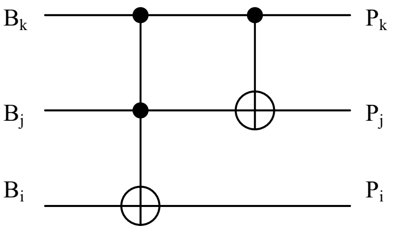

Figure 3: Peres gate $P_{i,j,k}$ built from Toffoli and Feynman gates.

Only half of all 4-qubit reversible circuits (even reversible circuits) can be realized by CNP (which is the alternative group A_{16}). The number $|A_{16}|$ is about 1 trillion (10^{13}). In our computer, the traditional algorithm with only forward search can only go 4 steps, and only about 1 million 4-qubit reversible circuits can be realized (see Table 3). When using enhanced bi-directional search, we set t=2 for time balance. The algorithm can use up to 8 steps. The number of reversible circuits with the minimum length becomes much larger. Denote Bf(k)=Af(k)-Af(k-1), the set of circuits with minimum length k. From the second step, the ratio of |Bf(j+1)|/|Bf(j)|, j=1,2,3, is 33.2, 29.9, 27.1 (see Table 2). Therefore, before the number of the represented circuits reaches half of the all even circuit number $|A_{16}|$, it is reasonable to assume that the ratio |Bf(j+1)|/|Bf(j)| is about 10% reduced forward. Based on this assumption, the estimation of the circuits with the minimum length represented by bi-direction method is 24*21*18*16 millions = $1.4*10^{11}$, about $1.4*10^5$ times more than using only the forward search. Using the enhanced bi-direction search, we calculate two more steps. $1.4*10^{11}*14*12 = 2.4*10^{13}$ bigger than half of $|A_{16}|$, which means over 50% of even 4-qubit reversible circuits can be optimally realized within the length of 10 (Table 3); after step 10, and the size of Bf(j) will decrease; and the maximum minimum length is less than 20.

Table 2: Number of circuits with minimum length *k*

k	1	2	3	4
|Af(k)|	41	1367	40967	1115774
|Bf(k)|	40	1326	39600	1074807
|Bf(k)|/|Bf(k-1)|		33.2	29.9	27.1

Table 3: Number of minimum synthesized circuits

	Forward search (one direction)		Enhanced Bi-direction search (s = 4, t = 2)	
Lib.	length	# circuits	length	# circuits
CNP	4	$1.0*10^6$	10	$> 5*10^{12}$

By using the enhanced bi-direction search, we present two 4-qubit reversible circuits with minimum lengths 8 and 9, respectively.

Example: Given f_1 as:

$P=AB+AC'+A'CD'$

$Q=A'C+B'CD'+BCD+BC'D'+BCD$

$R=BD+A'B'C+ABC+AC'D$,

$S=CD'+B'D$.

We get the minimum length of f_1 is 8 and:

$f_1=C_{B,C}*P_{C,B,D}*P_{B,D,C}*P_{C,B,D}*C_{A,D}*P_{C,A,D}*P_{A,B,C}*P_{B,A,D}$

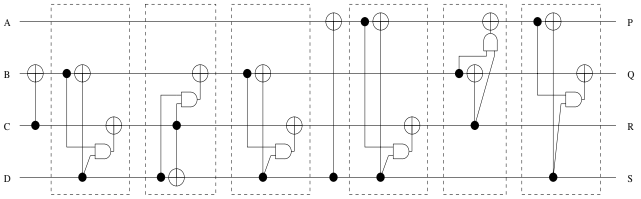

Figure 4: Optimal Realization of f_1

Given f_2 as:

P=AC'+AB'D+A'BC+A'CD'

Q=BD'+CD'+A'B'C+AB'C'D

R=BD+AC'D+ABC+A'B'C

S=AC+CD'+ABD+A'B'C'D.

The minimum length of f_2 is 9 and we have:

$f_2=P_{B,A,D}*P_{A,B,D}*P_{B,A,C}*P_{C,A,D}*C_{D,B}*P_{B,C,D}*P_{D,C,B}*P_{A,D,C}*C_{B,D}$

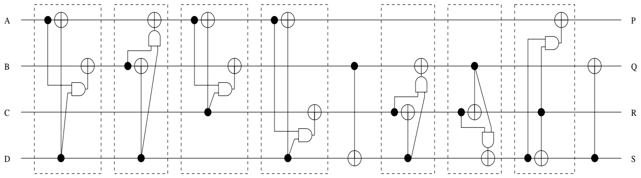

Figure 5: Optimal Realization of f_2

Our enhanced bi-directional search (EBS) is more efficient and powerful than a single forward search (FS). Not only the number of minimum synthesized circuits that can be handled by EBS is $5*10^6$ times bigger than that by FS, but also the speed of EBS is much faster than that of FS (see Table 4). Time is measured in seconds.

Table 4: Time of minimum synthesis

	Forward search		Enhanced Bi-direction search (s = 4, t = 2)	
Input circuits	length	time	length	time
g_1	4	188	4	1
g_2	exploded		5	4
g_3	exploded		6	5
f_1	exploded		8	508
f_2	exploded		9	1593

Where, the corresponding permutations of the input circuits g_1 , g_2 and g_3 are:

g_1 = (11,14,16)(12,13,15),

g_2 = (10,11,13,16)(12,15),

g_3 = (9,12,14)(11,16,13).

6. CONCLUSION

We presented an efficient bi-direction synthesis approach to the synthesis of binary reversible circuits using CNP library. Our proposed method outperforms all the existing methods in both speed and scalability. For the first time, we are able to find exact minimal solutions to some subset of 4-variable functions. The number of circuits that can be optimally synthesized is increased over $5*10^6$ times by our method comparing to standard search algorithms. The experimental results demonstrated the effectiveness of our approach. The method can be also used to invent new reversible 4-qubit gates to be next realized in quantum [13] and used in hierarchical heuristic-driven synthesis [5].

7. 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", *IEEE Trans. on Computer Aided Design of Integrated Circuits and Systems*, Vol.22, No. 6, June 2003, pp. 710-723.

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

[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*, pages 201-209, Trier, Germany, March 2003.

[6] S.S. Bullock and I.L. Markov. An arbitrary two-qubit computation in 23 elementary gates. *DAC*, pages 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. *DAC*, page 318-323, USA, June 2003.

[8] L. Storme et al. Group theoretical aspects of reversible logic gates. *Journal of Universal Computer Science*, 5 (1999), pages 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), pages 2855-2856.

[13] G. Yang, W. N. Hung, X. Song, and M. Perkowski. "Majority-based reversible logic gate", accepted by *Theoretical Computer Science*, in press, online available.

[14] X. Song, G. Yang, M. Perkowski, and Y. Wang. Algebraic characterization of reversible logic gates. To appear in *Theory of Computing Systems*, in press, online available.

[15] G. Yang, X. Song, W. N.N. Hung, M. A. Perkowski, "Fast Synthesis of Exact Minimal Reversible Circuits using Group Theory", *ASP DAC*, China, 2005, 1002-1005.