基于位运算的量子可逆逻辑电路快速综合算法*)

李志强[1,2] 陈汉武[1] 李文骜[1]

(东南大学计算机科学与工程学院 南京 210096)[1] (扬州大学信息工程学院 扬州 225009)[2]

摘 要 量子可逆逻辑电路是构建量子计算机的基本单元。本文结合可逆逻辑电路综合的多种算法,根据可逆逻辑电路综合的本质是置换问题,巧妙应用位运算构造高效完备的 Hash 函数,提出了基于 Hash 表的新颖高效的量子可逆逻辑电路综合算法,可使用多种量子门,以极高的效率生成最优的量子可逆逻辑电路,从理论上实现制造量子电路的成本最低。按照国际同行认可的 3 变量可逆函数测试标准,该算法不仅能够生成全部最优电路,而且运行速度远远超过其它算法。实验结果表明,该算法按最小长度标准综合电路的平均速度是目前最好结果的 69.8 倍。

关键词 量子电路优化,位运算,完备 Hash 函数,可逆逻辑电路

Speedy Algorithm for Synthesis of Quantum Reversible Logic Circuits Based on Bit Operation

LI Zhi-Qiang[1,2] CHEN Han-Wu[1] LI Wen-Qian[1]

(School of Computer Science & Engineering, Southeast University, Nanjing 210096)[1]

(College of Information Engineering, Yangzhou University, Yangzhou 225009)[2]

Abstract Quantum reversible logic circuits are basic elements of constructing quantum computer. This paper absorbs all kinds of ideas of synthesis of reversible logic circuits. Given that the essence of synthesizing reversible logic circuits is permutation, we use bit operation to construct the novel and perfect Hash function and present an efficient algorithm which can construct optimal quantum reversible logic circuits with various types of gates by using the Hash table and produce quantum circuits with minimal cost in theory. Judging by the internationally recognized reversible functions of three variables, the algorithm not only synthesizes all optimal reversible logic circuits, but also runs extremely faster than other ones. The experimental results show that the average speed of the algorithm which synthesizes circuit with minimum length is 69.8 times that of currently best result.

Keywords Quantum circuit optimization, Bit operation, Perfect Hash function, Reversible logic circuit

1 引言

量子计算机可等效一个量子图灵机,量子图灵机可等价一个量子逻辑电路,量子逻辑门的组合与级联是组成量子计算机的基本元素。所有量子逻辑门均可表示成复变空间酉矩阵,其输入与输出的比特数相等,也称可逆算子。量子逻辑门对输入比特进行确定的酉变换,得到输出比特。Deutsch[1]最早考虑用量子逻辑门构造量子计算机的问题,他发现几乎所有的三比特量子逻辑门都是通用逻辑门。该结果随后得到发展,最后 Deutsch[2]和 Lloyd[3]各自独立证明了几乎所有的二比特量子逻辑门都是通用的。实验上通常用一些具体的量子逻辑门构造量子计算机。Barenco[2]等人证明,一个二比特的异或门与对一比特进行任意操作的门可构成一个通用量子门集。迄今为止,虽然世界上还没有真正意义的量子计算机,但是世界主要经济发达国家都在制定战略性规划。正因为实现量子计算机技术困难重重,而它的实现必将为信息科学与通信技术带来革命性的突破,所以量子可逆逻辑电路的设计、优化方法的研究已越来越得到研究者的关注。量子可逆逻辑综合源于可逆计算机的研究。可逆逻辑已广泛应用在量子计算、低功耗 CMOS 电路、纳米技术、光计算,以及在数字信息处理、通信技术、计算机图像等领域的信息加密技术,因此可逆逻辑的研究已变得越来越重要。近 30 年来,人们已经提出了多种量子门,如 CNOT 门[4],Toffoli 门[5],Fredkin 门[6]等。如何使用指定量子门自动生成量子代价较小的量子电路,即制造量子电路的成本较低,进而实现制造量子计算机的成本较低,其本质是可逆逻辑综合问题,人们提出了一些算法,如 Shende[7]、Song[8]等人提出了一种 3 变量的综合方法;Iwama[10]等人提出了特定 CNOT 电路的综合规则;Miller[11,12]应用谱函数实现近似最优的可逆电路化简;Maslov[13]提出了先用真值表法构造量子电路,再用模板技术进行优化;Mishchenko[14]提出使用 ReedMuller 方法综合可逆逻辑电路;Gupta[15]给出了基于 ReedMuller 的启发式规则;Shende[9]将可逆逻辑电路综合简化为置换问题,并提出了性能较好的递归算法;Yang[16]在此基础上将可逆逻辑电路综合进一步抽象为群论问题,并设计了基于群论 GAP 软件的量子电路综合算法,其性能远远超过其它算法。然而目前人们还没有找到通用高效的算法,特别针对多变量的量子电路,这是量子电路中急需解决的重要问题之一,因为这不仅可以降低制造量子电路的成本,而且能优化许多量子计算算法。Song 等人提出的穷举算法太慢;Miller 等人提出的模板优化技术速度较慢,且

*)国家自然科学基金(60572071)、国家自然科学基金会重大研究计划(90412014)、江苏省自然科学基金(BK2005053)(BM2006504)、江苏省高校自然科学基金资助项目(06KJB520137)。李志强 博士生,讲师,主要研究方向:量子计算、可逆电路综合与测试;陈汉武 博士,教授,博士生导师,主要研究方向:量子计算、信息论;李文骜 硕士,主要研究方向:量子计算模拟。

生成的电路难以达到最优;Iwama 提出的 CNOT 电路的综合规则比较繁琐,且对电路有特别要求;Mishchenko、Gupta 给出了基于 ReedMuller 技术的启发式算法,算法性能有所改善,但不具有通用性;Shende 应用置换对算法,性能有本质性的提高,Yang 应用群论将算法,性能远远超过其它算法,但该算法完全依赖 GAP 群论软件,没有提炼出核心算法,算法性能还可以大幅度提高。本文以此为基础,巧妙运用位操作构造完备 Hash 函数,极大地提高了算法性能,按最小长度标准综合量子电路的平均速度是目前最好结果[16]的 69.8 倍。

2 量子可逆逻辑电路的基本概念

利用微观粒子状态表示的信息称为量子信息,量子信息的基本单位是量子比特(qubit)。与经典信息不同,量子比特能够以叠加态的形式存在。任何量子比特均可由一个二元向量形式表示为$|\phi\rangle=\alpha|0\rangle+\beta|1|$,其中 α 和 β 为复数,满足归一化条件$|\alpha|^2+|\beta|^2=1$。量子逻辑门是处理量子信息的基本单元,它的级联构成量子电路。量子电路必须是可逆的,即量子信息的动态过程在复向量空间上必须保持正交变换。在量子计算中,一个量子逻辑门对应一个幺正变换。根据输入输出的对数,逻辑门可分为单量子比特门与多量子比特门。

定义 1 Toffoli 量子门,记为 $TOF(C;F)$,其中输入变量集合 $In=\{x_1,x_2,\cdots,x_n\}$,控制端集合 $C=\{x_{i_1},x_{i_2},\cdots,x_{i_{n-1}}\}$,受控端集合 $T=\{x_{i_n}\}$,且 $C\cap T=\varnothing$,$C\cup T\subset In$。输出变量集合映射为$\{x_1,x_{i_2},\cdots,x_{i_n-1},x_{i_n}\oplus\prod_{k=1}^{n-1}x_{i_k},x_{i_n+1},\cdots,x_n\}$。若$\exists m\in\{1,2,\cdots,n-1\}$,$x_{i_m}=0\to\prod_{k=1}^{n-1}x_{i_k}=0$,受控端 x_{i_n} 的输出为$x_{i_n}\oplus\prod_{k=1}^{n-1}x_{i_k}=x_{i_k}\oplus0=x_{i_n}$;若$\forall m\in\{1,2,\cdots,n-1\}$,$x_{i_m}=1\to\prod_{k=1}^{n-1}x_{i_k}=1$,受控端 x_{i_n} 的输出为$x_{i_n}\oplus\prod_{k=1}^{n-1}x_{i_k}=x_{i_k}\oplus1=\overline{x_{i_n}}$。如图 1 所示,当 $n=1$ 时 $C=\varnothing$,$TOF(x_{i_1})$ 为非门(NOT),简称 N;当 $n=2$ 时,$C=\{x_{i_1}\}$,$TOF(x_{i_1};x_{i_2})$ 为控制非门(CNOT),简称 C;当 $k=3$ 时,$C=\{x_{i_1},x_{i_2}\}$,$TOF\{x_{i_1}x_{i_2};x_{i_3}\}$为标准 Toffoli 门,简称 T;其中只有非门为单量子比特门,其它均为多量子比特门。

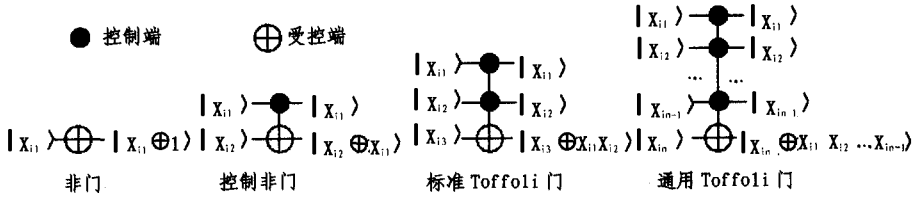

图 1 量子逻辑门

定义 2 控制交换门,即 Fredkin 门,记为 $FRE(C;T)$,简称 F,其中控制端集合 $C=\{x_{i_1},x_{i_2},\cdots,x_{i_{n-2}}\}$,受控端集合 $T=\{x_{i_{n-1}},x_{i_n}\}$,输入变量集合 $In=\{x_1,x_2,\cdots,x_{i_{n-1}-1},x_{i_{n-1}},x_{i_{n-1}+1},\cdots,x_{i_n-1},x_{i_n},x_{i_n+1},\cdots,x_n\}$,且 $C\cap T=\varnothing$,$C\cup T\subset In$。若$\exists m\in\{1,2,\cdots,n-2\}$,$x_{i_m}=0\to\prod_{k=1}^{n-2}x_{i_k}=0$,输出变量集合映射为 In;若$\forall m\in\{1,2,\cdots,n-2\}$,$x_{i_m}=1\to\prod_{k=1}^{n-2}x_{i_k}=1$,输出变量集合映射为$\{x_1,x_2,\cdots,x_{i_{n-1}},x_{i_n},x_{i_{n-1}+1},\cdots x_{i_{n-1}},x_{i_{n-1}},x_{i_n+1},\cdots,x_n\}$。

定义 3 Peres 门,记为 $Pe(x_1,x_2,x_3)$,简称 P,其中输入变量集合 $In=\{x_1,x_2,x_3\}$,输出变量集合映射为$\{x_1,x_2\oplus x_2,x_3\oplus x_1x_2\}$,其功能相当于量子门 $TOF(x_1x_2;x_3)$ 与 $TOF(x_1;x_2)$ 级联。

定义 4 设 $B=\{0,1\}$,有 n 个输入与 n 个输出变量的布尔函数$f:(x_1,x_2,\cdots,x_n)\to\{y_1,y_2,\cdots,y_n\}$,即 $f:B^n\to B^n$,$\langle x_n,x_{n-1},\cdots,x_1\rangle\in B^n$是输入向量,$\langle y_n,y_{n-1},\cdots,y_1\rangle\in B^n$ 是输出向量,布尔逻辑电路 f 是可逆的当且仅当 f 是双射,即既是单射又是满射。一个布尔逻辑电路有 n 个输入和 n 个输出,称为 $n\times n$ 的可逆逻辑电路,也称 n 变量可逆逻辑电路。为方便处理,通常将输入、输出向量表示成整数$\langle x_n,\cdots,x_2,x_1\rangle_2$ 与$\langle y_n,\cdots,y_2,y_1\rangle_2$,其中,$x_i,y_i,1\leqslant i\leqslant n$ 分别表示第 i 个输入与第 i 个输出变量,$\langle B_n,B_{n-1},\cdots,B_1\rangle_2=\sum_{i=1}^{n}B_i\cdot2^{i-1}$。

定义 5 设 M 为有限集,$\sigma:M\to M$ 为双射,则称 σ 为 M 上的置换,设 M 中有 n 个数,即 $|M|=n$,所以称 σ 为 n 次置换。置换常表示为$\sigma=\begin{pmatrix}0&1&\cdots&n-1\\p_0&p_1&\cdots&p_{n-1}\end{pmatrix}$,$M$ 上的恒等函数为$\sigma=\begin{pmatrix}0&1&\cdots&n-1\\0&1&\cdots&n-1\end{pmatrix}$,显然量子电路不存量子门的置换。可逆函数可用真值表描述,也可用整数集合$\{0,1,\cdots,2^n-1\}$的置换表示。图 2 是一个 3 变量的量子可逆逻辑电路,用真值表描述见表 1,用置换表示为 $\sigma=\begin{pmatrix}0&1&2&3&4&5&6&7\\2&6&0&1&7&3&5&4\end{pmatrix}$,即输入分别为 0,1,…,7 时,输出分别为 $p_0=2,p_1=6,\cdots,p_7=4$。量子电路由若干个量子门级联而成,则量子电路实现的置换等价于这些量子门的置换的乘积。如图 2 中,NOT 门的置换为 $\sigma_1=\begin{pmatrix}0&1&2&3&4&5&6&7\\2&3&0&1&6&7&4&5\end{pmatrix}$,Toffoli 门的置换为 $\sigma_2=\begin{pmatrix}0&1&2&3&4&5&6&7\\0&1&2&7&4&5&6&3\end{pmatrix}$,CNOT 门的置换为 $\sigma_3=\begin{pmatrix}0&1&2&3&4&5&6&7\\0&1&2&3&5&4&7&6\end{pmatrix}$,显然 $\sigma\equiv\sigma_1\circ\sigma_2\circ\sigma_3$。

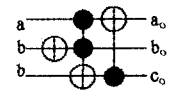

图 2 量子可逆逻辑电路

表 1 真值表

输入		输出	
$(x_3x_2x_1)_2$	$x_3x_2x_1$	$(y_3y_2y_2)_2$	$y_3y_2y_1$
$(c\,b\,a)_2$	$c\,b\,a$	$(c_o\,b_o\,a_o)_2$	$c_o\,b_o\,a_o$
0	000	2	010
1	001	6	110
2	010	0	000
3	011	1	001
4	100	7	111
5	101	3	011
6	110	5	101
7	111	4	100

定义 6 量子门库 L_n 表示 $n\times n$ 量子门的集合,它用于综合 $n\times n$ 的可逆逻辑电路,简写为 L,用 $T(L)$表示库 L 能综

合的所有 $n\times n$ 量子可逆逻辑电路的集合。

定义 7 $\min l(a)$表示 $T(L)$中任意量子电路 a 的最小长度，即该电路至少从量子门库 L 中由重复选择 $\min l(a)$个量子门级联而成，$\max l(T(L))$表示 $T(L)$中所有量子电路的最小长度的最大值。

2 量子可逆逻辑电路综合

由定义 5 可知，量子门的功能本质上是实现数据的某种置换。量子电路是若干个量子门的级联，因此量子电路的功能本质上是若干数据置换的叠加，即置换的乘积，最终也是实现数据的某种置换。研究表明，直接综合等于或大于 4 变量的量子电路是非常困难的，但可将它们分解成若干个 3 变量的量子电路综合的问题。本文方法主要针对 3 变量的量子电路综合问题，利用位操作，巧妙构造完备 Hash 表，以目前国际文献中最快的速度综合量子电路。

2.1 Hash 表及 Hash 函数的构造

如前所述，量子可逆逻辑电路的功能，本质上是实现数据的某种置换。若量子电路有 n 条输入、输出线，根据定义 4 可知，全部输入、输出向量表示成整数为$\{(\underbrace{00\cdots00}_{n})_2,(\underbrace{00\cdots1}_{n})_2,\cdots(\underbrace{11\cdots1}_{n})_2\}$，用十进制数表示分别为$(0,1,\cdots 2^n-1)$。为方便处理，一般都用十进制数间接表示 n 个输入、输出变量的值。设输入值分别为$(0,1,\cdots,2^n-1)$，输出值分别为$(p_0,p_1,\cdots,p_{2^n-1})$，则对应的置换为 $\sigma=\begin{pmatrix}0 & 1 & \cdots & 2^n-1\\ p_0 & p_1 & \cdots & p_{2^n-1}\end{pmatrix}$，显然这样的置换共有 $2^n!$ 种。为生成最较优的电路，电路的综合算法中需要经常判断当前置换是否已在历史置换数据中存在。现有的算法中，有些通过顺序查找判断。文[16]通过 GAP 群论软件判断，因置换群中不允许存在 0 元素，所以算法为直接使用 GAP 软件而将每个置换数据加 1，这些算法的时间复杂度都非常高。为此，本文运用 Hash 算法仅需一步就能判断，且构造的 Hash 函数计算简单，从而实现数据的快速查找与判断，并且减少 Hash 表的长度，降低算法的空间复杂度。构造 Hash 函数的方法如下：

已知$(p_0,p_1,\cdots,p_{2^n-1})$是$(0,1,\cdots,2^n-1)$的一个置换，显然 $p_{2^n-1}=\{0,1,\cdots,2^n-1\}-\{P_0,P_1,\cdots,P_{2^n-1}\}$，可得 P_{2^n-1}可由$\{P_0,P_1,\cdots,P_{2^n-2}\}$唯一确定，所以$(P_0,P_1,\cdots,P_{2^n-2},P_{2^n-1})$与$(p_0,p_1,\cdots,p_{2^n-2})$一一对应。而 $p_i\in\{0,1,\cdots,2^n-1\}$，$0\leqslant i\leqslant 2^n-2$，因此 p_i 可用 n 位二进制数表示。将这些二进制数依次降序排列，形成长度为 $n(2^n-1)$的二进制数。令该二进制数对应的十进制数是 Hash 地址，则 Hash 函数为 $H(p_0,p_1,\cdots,p_{2^n-2},p_{2^n-1})=\sum_{j=0}^{2^n-2}p_j\cdot 2^{j\cdot n}$。

该 Hash 表的地址最大值 MAXH 与最小值 MINH 分别为：

$$\mathrm{MAX}H=\max(H(p_0p_1\cdots p_{2^n-1}))=\max(\sum_{j=0}^{2^n-2}p_j\cdot 2^{j\cdot n})=\sum_{j=0}^{2^n-2}\max(p_j)\cdot 2^{j\cdot n}=\sum_{j=0}^{2^n-2}(j+1)\cdot 2^{j\cdot n}<2^{(2^n-1)}\cdot n$$

$$\mathrm{MIN}H=\min(H(p_0,p_1,\cdots,p_{2^n-1}))=\min(\sum_{j=0}^{2^n-2}p_j\cdot 2^{j\cdot n})=\sum_{j=0}^{2^n-2}\min(p_j)\cdot 2^{j\cdot n}=\sum_{j=0}^{2^n-2}(2^n-2-j)\cdot 2^{j\cdot n}>0$$

所以 Hash 表的长度为 $\mathrm{MAX}H-\mathrm{MIN}H+1=\sum_{j=0}^{2^n-2}(2j+3-2^n)\cdot 2^{j\cdot n}<2^{(2^n-1)\cdot n}$，即空间复杂度为 $O(2^{2^n-1}\cdot n)$。

为节省内存空间，Hash 表中的节点可用位域表示，其中数据项 *gate* 表示节点对应的量子门，也是对应量子电路在 Hash 表中存在的标志，数据项 *prev* 是与当前量子门级联的前面量子门在 Hash 表中的地址。

当 $n=3$，输入、输出值分别为$\{0,1,\cdots,2^3-1\}$，设输入值分别为$(0,1,\cdots,7)$，输出值分别为$(p_0,p_1,\cdots,p_7)$，即$(0,1,\cdots,7)$的某种置换，而$(0,1,\cdots,2^3-1)$共有 $2^3!=40320$ 种不同的置换。已知$(0,1,\cdots,7)$的任意置换是由 8 个 0～7 的整数组成，由 Hash 函数 H 可知，只要保存前面 7 个数，各数用 3 位二进制表示，共用 21 位二进制数表示，则 $\mathrm{MIN}H=(\underbrace{000}_{p_6}\underbrace{001}_{p_5}\underbrace{010}_{p_4}\underbrace{011}_{p_3}\underbrace{100}_{p_2}\underbrace{101}_{p_1}\underbrace{110}_{p_0})_2$，$\mathrm{MAX}H=(\underbrace{111}_{p_6}\underbrace{110}_{p_5}\underbrace{101}_{p_4}\underbrace{100}_{p_3}\underbrace{011}_{p_2}\underbrace{010}_{p_1}\underbrace{001}_{p_0})_2$，即分别为 42798，2054353，因此 Hash 表中仅需 2011556 个节点。已知每个节点占用 4 个字节，则此表共占用内存 7.67MB。

2.2 完备 Hash 函数证明

函数 H 是完备的 Hash 函数(Perfect Hash Function，PHF)，又称无冲突函数。

证明：设有任意两个不相同的置换$(P_0,P_1,\cdots,P_{2^n-2},P_{2^n-1})$、$(Q_0,Q_1,\cdots,Q_{2^n-2},Q_{2^n-1})$，可得如下两个结论：

1) 这两个置换的前 2^n-2 项数据一定不完全相同。

假设前 2^n-2 项数据完全相同，即$(P_0,P_1,\cdots,P_{2^n-2})=(Q_0,Q_1,\cdots,Q_{2^n-2})$，而由 2.1 可知 $P_{2^n-1}=\{0,1,\cdots,2^n-1\}-\{P_0,P_1,\cdots,P_{2^n-2}\}$，$Q_{2^n-1}=\{0,1,\cdots,2^n-1\}-\{Q_0,Q_1,\cdots,Q_{2^n-2}\}$，可得 P_{2^n-1},Q_{2^n-1}，因此$(P_0,P_1,\cdots,P_{2^n-1},P_{2^n-1})=(Q_0,Q_1,\cdots,Q_{2^n-2}Q_{2^n-1})$，即这两个置换相同，与条件不相符，则假设不成立，结论 1 得证。

2) Hash 函数 H 将这两个置换映射为不同的 Hash 地址。

显然置换中的数据都可用长度 n 的二进制数表示，由 Hash 函数 H 的定义可知，该函数是将置换的前 2^n-2 项数据，分别用长度 n 的二进制数表示，并依次降序排列，生成长度为$(2^n-2)n$ 的二进制数，其对应的十进制数便是 Hash 地址。由结论 1 可知，这两个置换的前 2^n-2 项数据一定不完全相同，可得这两个置换生成长度为$(2^n-2)n$ 的二进制数也一定不相同，因此其对应的十进制数即 Hash 地址也一定不相同。

从结论 2 可知，$H(P_0,P_1,\cdots,P_{2^n-2},P_{2^n-1})\neq H(Q_0,Q_1,\cdots,Q_{2^n-2},Q_{2^n-1})$，因此 Hash 函数是完备 Hash 函数。 □

Hash 函数 H 将每个不同的置换都映射为不同的 Hash 地址，有效避免了所有冲突，平均与最坏情形的时间复杂度都为 O(1)，它优于任何一个已知的检索算法，确实是完美、完备的。当然，这样的 Hash 函数是很难获得的，而且置换与 Hash 地址之间相互转换的效率较高，这些都是本文算法极快的根本原因。

2.3 计算 Hash 函数的快速算法

在综合全部 n 变量的量子电路过程中，若量子门库中的量子门数为 m，则需要计算 $m2^n!$ 次 H 函数。

证明：已知 n 变量的量子电路共有 $2^n!$ 种置换，每种置换都要从量子门库中取出 m 个量子门试探，而每次试探都要计算一次 H 函数，以判断当前电路是否最优，因此计算 H 函数的总数为 $m2^n!$ 次。 □

为进一步提高综合算法的整体性能，H 函数的计算效率

非常重要。本文运用 C++ 语言的位操作，将量子电路的置换快速转换成 Hash 地址，其时间复杂度仅为 $O(2^n)$。

算法 1 计算 Hash 函数的算法 PtoAddr

输入：数据置换($P_0, P_1, \cdots, P_{2^n-2}, P_{2^n-1}$)
输出：该置换在 Hash 表中的地址
1. for i=0 to 2^n-2
2. ibitlen[i]=$i \cdot n$
3. end for
4. ireet=0
5. for i=0 to 2^n-2
6. iret=iret|(p_i<<ibitlen[i])
7. end for
8. return iret

算法 PtoAddr 是将任意 n 变量的量子可逆逻辑电路输出的置换转换成 Hash 地址，第 1～3 步初始化全局数组 *ibitlen*，分别存放了 $P_0, P_1, \cdots, P_{2^n-2}$ 的二进制数在 Hash 地址对应的二进制数的位置。当 n 确定时，这些位置是固定不变的，即数组中的值保持不变，因此只需第一次调用该算法时运行这 3 步；第 4～7 步是将 p_i 向左位移 *ibitlen*[i]位，再与 *iret* 位或运算；第 8 步将获得 Hash 地址 *iret*。

2.4 最小长度整体综合算法

设量子电路有 n 条量子线，量子门库中有 m 个不同的量子门，即 m 个不同的置换规则。有重复地选择若干量子门级联，构成的电路分别实现全部不同的置换要求得到的每个电路即每个置换，选择的量子门数最少。由 2.1 可知，每个电路的置换对应的 Hash 地址 $h \in \{MINH, MINH+1, \cdots, MAXH\}$，其中包括了全部 $2^n!$ 个不同电路对应的 Hash 地址，因此从仅需 0 个量子门的电路开始，依次试探 m 个不同的量子门，即 m 个不同的置换规则。将电路的输出值对应的置换生成 Hash 地址，并写到 Hash 表中。显然，0 个量子门一定是最优的。设已有所有长度为 l 的最优电路，则所有长度为 $l+1$ 的最优电路的生成方法是将长度为 l 的最优电路的后面分别试探 m 个不同的量子门，即长度为 l 的最优电路的置换分别乘以 m 个不同量子门的置换，并将得到的置换转变为 Hash 地址，判断 Hash 表的该位置是否已有数据。如果没有，则写入数据，否则说明在此之前一定存在功能相同即置换相同，且长度不大于当前量子电路的长度，这是因为算法中电路是按长度从小到大依次生成，可得当前电路长度一定大于或等于已有相同功能的电路，因此要去除。依次类推，直至电路试探生成的全部量子电路对应的 Hash 地址所指的位置都存在数据，即没有生成新功能的量子电路。

算法 2 最小长度量子电路整体综合算法 QML

输入：量子门库 L
输出：$j, n[0\cdots j]$, hash[MIN$H\cdots$MAXH]
1. haxh=$\emptyset$
2. b=(0,1,$\cdots$,2^n-1), $B[0]=\{b\}$
3. j=0, $n[j]$=1, haxh[$H(b)$]. prev=-1
4. while $n[j]\neq 0$
5. $B[j+1]=\emptyset$
6. for all b in $B[J]$
7. $ikey\_prev=H(b)$
8. for all a in $T(L)$
9. $p=b^* a, ikey=H(p)$
10. if hash[$ikey$]. $gate$ is null then
11. hash[$ikey$]. $gate$=a, haxh[$ikey$]. $prev=ikey\_prev$
12. $B[j+1]=B[j+1]\cup\{p\}$
13. end if
14. end for
15. end for
16. $n[j+1]=|B[j+1]|, j=j+1$
17. end while

算法 QML 将生成全部长度最小的 n 变量的最优量子电路。统计出各个长度的电路总数，存入数组 n 中，算法返回的 j 表示这些电路的最大长度，即 $j=\max l(T(L))$。第 2 步，b 表示电路中没有任何量子门，实现恒等置换；第 3 步，j=0 表示电路有 0 个门，$n[j]$=1 表示长度为 j 的电路总数为 1，数据项 $prev=-1$ 表示该量子门前面没有量子门；算法第 4 步表示如果还存在长度为 j 的最优电路，则试求长度为 $j+1$ 的最优电路；第 5 步置长度为 $j+1$ 最优电路的置换构成的集合 $B[j+1]$为空；第 6 步依次取出长度为 j 的最优电路对应的置换 a；第 8～14 步是在长度为 j 的最优电路 b 后面追加量子门库中的量子门，即将置换 b 依次与量子门库 L 中的每个量子门对应的置换 a 乘积，生成新的置换 p；其中，第 9 步中 $b*a$ 表示置换 b 经过置换 a 后，得到的新置换，设置换 b、a 分别用数组 $b[2^n]$、$a[2^n]$表示，则 $b*a=(b[1],b[2],\cdots,b[2^n])*(a[1],a[2],\cdots,a[2^n])=(a[b[1]],a[b[2]],\cdots,a[b[2^n]])$；第 10 步置换 p 对应的 Hash 地址为 *ikey*，判断在 Hash 表的 *ikey* 位置是否存在数据。若不存在，第 11 步将 Hash 表的 *ikey* 位置的数据域 *gate* 设置为当前追加的量子门 a，并将数据域 *prev* 指向电路 b 对应的 Hash 表中的位置 *ikey_prev*；第 11 步将置换 p 加入到长度为 $j+1$ 的最优电路集合中；第 16 步 $B[k]=\{a \mid a\in T(L)\wedge \min l(a)=k\}$，因此集合 $B[k]$的长度是长度为 j 的最优量子电路的总数，存入 $n[k]$中。

2.5 基于算法 QML 综合具体量子可逆逻辑电路的算法

本文在实现前面整体综合算法时，通过 Hash 表中节点的数据项 *prev* 在逻辑上已构建了一棵多叉树，其中包含了全部 n 变量的最优量子电路。根据所求量子电路的置换计算出 Hash 地址为 *ikey*，找到树中对应节点 *hash*[*ikey*]，该节点的 *gate* 数据项就是这个量子电路的最后一个量子门。再根据该节点的 *prev* 数据项，找到它的父节点，而父节点的 *gate* 数据项就是与该门级联的前一个量子门。依次类推，直至达到根节点，即 *ikey* 值等于 -1，为结束标志，将经过的全部量子门逆序级联，从而生成所求量子可逆逻辑电路。

算法 3 根据算法 QML 生成任意 n 变量的量子电路算法 QMR

输入：量子门库 L，所求量子电路的置换 p
输出：满足要求的长度最优量子电路的量子门序列
1. 仅第一次要计算 QML(L)，生成 hash[MINH..MAXH]
2. $ikey=H(p), j=0$
3. while $ikey\neq -1$ do
4. $Gate[j]$=hash[$ikey$] . $gate$
5. $ikey$=hash[$ikey$]. $prev$
6. $j=j+1$
7. end while
8. return $Gate[j-1], Gate[j-2], \cdots, Gate[0]$

算法 3 是在算法 2 运行完毕的基础上，生成任意置换为 p 的长度最优的量子电路。第 2 步计算 p 的 Hash 地址，若为 -1，则量子电路中没有量子门；否则在第 3～7 步从多叉树的 $H(p)$节点出发依次访问其父子节点，直至到达根节点，即节点的 prev=-1；第 8 步将访问路径中节点对应的量子门按访问顺序逆序排列，并生成所求长度最小的量子电路。

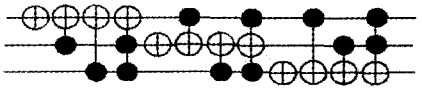

图 3 全部 3 变量的 NCT 量子门

2.6 算法复杂性分析

本文算法与其它算法相比具有显著的优势。设量子电路的量子线数为 n，求 Hash 地址的算法的时间复杂度为 $O(2^n)$。若使用 NCT 量子门库，共有 $(n^3-n^2+2n)/2$ 种不同的量子门，全部可逆电路共有 $2^n!$ 种，而每个电路都依次试探

全部量子门，则共试探 $2^n!\ (n^3-n^2+2n)/2$ 次。如 n 等于 3，量子门共有如图 3 的 12 种，本文的最小长度综合全部 $2^n!$ 个量子电路的算法时间复杂度为 $O(2^n 2^n!\ (n^3-n^2+2n)/2)=O(2^{n-1}2^n!\ (n^3-n^2+2n))$，则最小长度综合每个量子电路的算法平均时间复杂度为 $O(2^{n-1}(n^3-n^2+2n))$。在此基础上，综合长度 l 的最优可逆电路的算法 3 中仅需访问 $l+1$ 个节点。当 n 较小时，该时间复杂度并不很大，事实上其它算法都非常复杂，以至于没有一篇文献能给出其算法复杂度。本文算法使用的 Hash 表较长，空间复杂性较高 $O(2^{(2^n-1))\cdot n})$，因此该算法是用空间换取时间，在综合大电路时可引入空间压缩技术，确保算法高速稳定运行。

本文算法使用 C++ 编程，并运用了多种编程技术，如巧妙运用位操作，提高 Hash 地址与数据置换序列之间的转换效率；缓存常用的中间计算结果，减少重复计算相同的数据，增强内存的复用率，避免反复申请与释放内存；借鉴群论思想，快速递推生成全部最优量子电路；为显示本算法的优越性，做了大量实验。

3 实验结果与分析

本文采用国际同行认可的 3 变量可逆函数测试标准的实验，生成共 $2^3!=8!=40320$ 个可逆逻辑电路。这个实验的目的是用较短的时间找到全部量子代价尽可能小的电路。实验中，本文采用多种量子门库与最小长度标准，快速生成全部最优电路。在量子电路综合的许多算法中，有些应用启发式规则或模板技术，但不能得到最优解；有些利用穷举法得到最优解，但速度很慢，如本领域的权威 Miller 教授的最新文献公布的情况是，在 Sun Blade 1000 750MHz 电脑上，应用迭代算法，平均门数为 6.38，历时 33h；增加模板等优化技术，达到近似最优，历时却有 96h；文[16]巧妙将量子电路综合问题抽象为群论问题，其算法性能与其它算法相比遥遥领先。为更好地与文[16]比较，本实验的电脑为联想奔腾Ⅲ 667MHz 64M，而 Yang 实验的电脑为奔腾Ⅲ 850MHz，比本实验的电脑配置略高些。QML 算法的运行结果见表 2。实验数据表明，本文的算法简洁高效，便于理解与应用，不仅得到全部最优解，而且运行速度极快，按最小长度标准综合的平均速度是目前最好结果[16]的 69.8 倍。

表 2 最小长度为 K 的量子电路的数量

最小长度 k	NC	NCT	NCP	NCF	NCPT	NCTF	NCPF	NCTPF
0	1	1	1	1	1	1	1	1
1	9	12	15	12	18	15	18	21
2	51	102	174	101	228	143	248	281
3	187	625	1528	676	1993	1006	2356	2551
4	393	2780	8968	3413	10503	5021	12797	13181
5	474	8921	23534	11378	23204	15083	22794	22323
6	215	17049	6100	17970	4373	17261	2106	1962
7	14	10253	0	6739	0	1790	0	0
8	0	577	0	30	0	0	0	0
总数	1344	40320	40320	40320	40320	40320	40320	40320
平均门数	4.47	5.87	4.84	5.66	4.73	5.33	4.6	4.57
最大长度	7	8	6	8	6	7	6	6
时间(s)	0.03	0.14	0.17	0.14	0.19	0.17	0.19	0.22
文[16]时间	无	12	10	13	10	12	11	无
提高倍数	无	85.71	58.82	92.86	52.63	70.59	57.89	无

结束语 本文根据可逆逻辑电路综合本质就是置换问题的基本思想，巧妙运用位操作构造高效完备的 Hash 函数，提高了 Hash 地址与置换之间的转换效率，提出了一种新颖高效的量子电路综合算法，使用多种量子门，采用最小长度标准，以极高的效率生成最优的量子可逆逻辑电路。如何以较高效率综合大规模量子电路，在本文的基础上，还需要引入新的技术与方法，这是笔者所在实验室正在研究的重要课题之一。

参 考 文 献

1 Deutsch D. Quantum computational networks. In: Proceedings of the Royal Society, London A, 1985,425:73～90

2 Deutsch D, Barenco A, Ekert A. Universality in quantum computation. In: Proceedings of the Royal Society, London A, 1995, 449:669～677

3 Lloyd S. Almost any quantum logic gate is universal. Physical Review Letters,1995, 75(2):346

4 Feynman R. Quantum mechanical computers. Optic News, 1985. 11～20

5 Toffoli T. Reversible computing. In: de Bakker J W, Van Leeuwen J, eds. Automata, Languages and Programming. New York: Springer, 1980

6 Fredkin E, Toffoli T. Conservative logic. International Journal of Theoretical Physics,1982,21:219～253

7 Shende V V, Prasad A K, Markov I L, et al. Reversible logic circuit synthesis. In: Proceedings of the International Conference on Computer-Aided Design, California, 2002. 125～132

8 Song X Y, Yang G W, Perkowski M, et al. Algebraic characteristics of reversible gates. Theory of Computing Systems, 2004. 311～319

9 Shende V V, Prasad A K, Markov I L, et al. Synthesis of reversible logic circuits. In: IEEE Transactions on Circuits and Systems-I, 2003, 22(6):723～729

10 Iwama K, Kambayashi Y, Yamashita S. Transformation rules for designing CNOT-based quantum circuits. In: Proceedings of Design Automation Conference, New Orleans, 2002, 28(4):419～424

11 Miller D M, Maslov D, Gueck G W. Spectral and two-place decomposition techniques in reversible logic. In: Proceedings of the 45th IEEE International Midwest Symposium on Circuits and Systems, Tulsa, 2002. 493～496

12 Miller D M. A transformation based algorithm for reversible logic synthesis. In: Proceedings of the International Conference on Computer-Aided Design, California, 2003. 318～323

13 Maslov D, Dueck G W, Miller D M. Toffoli network synthesis with templates. IEEE Transactions on Circuits and Systems-I, 2005, 24(6):807～817

14 Mishchenko A, Perkowski M. Logic synthesis of reversible wave cascades. In: Proceedings of 11th IEEE International Workshop on Logic Synthesis, New Orleans, 2002. 197～202

15 Gupta P, Agrawal A, Jha N K. An Algorithm for Synthesis of Reversible Logic Circuits. IEEE Transactions on Circuits and Systems-I, 2006,25(11):807～817

16 Yang G W, Song X Y, Hung W N N, Perkowski MA. Fast synthesis of exact minimal reversible circuits using group theory. In: Proceedings of IEEE ASP-DAC 2005, Shanghai, China, 2005, 2: 18～21