四量子可逆逻辑电路快速综合算法

李志强[1,2],陈汉武[1],徐宝文[1],肖芳英[1],薛希玲[1]
(1. 东南大学计算机科学与工程学院,江苏南京 211189;2. 扬州大学信息工程学院,江苏扬州 225009)

摘 要: 量子可逆逻辑电路综合是以较小量子代价自动构造所求量子可逆逻辑电路.本文提出了一种新颖高效的 4 量子电路综合算法,巧妙构造置换的最短编码,通过对量子电路进行特定拓扑变换,无损压缩 n 量子最优电路占用内存空间近 $2\times n!$ 倍,通过对已生成最优电路的双向级联,可使用多种量子门,采用最小长度标准,以极高效率生成较长的 4 量子电路,如率先生成基于控制非门、非门、Toffoli 门库的全部前 8 层共 3120218828 个电路,还可快速综合任意长度不超过 16 的最优电路,并对 4 量子标准测试电路进行快速且全面的优化.

关键词: 4 量子; 可逆逻辑综合; 最短编码; 拓扑压缩; 量子计算
中图分类号: TP387 **文献标识码**: A **文章编号**: 0372-2112 (2008) 11-2081-09

Fast Algorithms for 4-qubit Reversible Logic Circuits Synthesis

LI Zhi-qiang[1,2], CHEN Han-wu[1], XU Bao-wen[1], XIAO Fang-ying[1], XUE Xi-ling[1]
(1. *School of Computer Science and Engineering, Southeast University, Nanjing, Jiangsu* 210096, *China*;
2. *College of Information Engineering, Yangzhou University, Yangzhou, Jiangsu* 225009, *China*)

Abstract: Synthesis of quantum reversible logic circuits means to automatically construct desired quantum reversible logic circuit with minimal quantum cost. We present a novel and efficient algorithm which can construct almost all optimal 4-qubit reversible logic circuits with various types of gates and minimum length cost based on constructing the shortest coding and the specific topological compression, whose lossless compression ratios of the space of n-qubit circuits is near $2\times n!$. We firstly have created all 3120218828 optimal 4-qubit circuits whose length is less than 9 for the Controlled-Not gate, NOT gate and Toffoli gate library, and our method can achieve 16 steps through cascading created circuits. Our algorithm can not only synthesizes all the 4-qubit benchmark circuits, but also runs extremely fast.

Key words: 4-qubit; reversible logic synthesis; shortest coding; topological compression; quantum computing

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[1],因此可以通过量子逻辑门的级联与组合构成量子计算机.量子可逆逻辑综合源于可逆计算机的研究,现已广泛应用在量子计算、低功耗 CMOS 电路、纳米技术、光计算和信息加密等领域中,因此可逆逻辑的研究已变得越来越重要.近 30 年来,人们已提出了多种量子门,如 CNOT 门, Toffoli 门, Fredkin 门[2]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,从而实现制造量子计算机的成本较低,其本质就是可逆逻辑综合问题,为此人们提出了一些综合算法,如 Song[3]等人给出了可逆逻辑门的代数特征; Maslov[4]提出了先用真值表法构造量子电路,再用模板技术优化电路; W Q Li[5]给出了通用的模板构造与优化算法; Gupta[6]给出了基于 ReedMuller 的启发式规则; Shende[7]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[8]在此基础上将可逆逻辑电路综合进一步抽象为群论问题,算法性能远远超过其他算法;然而人们还没有找到高效综合大型量子电路的算法.目前大部分算法只能综合 3 量子电路,在综合 4 量子电路时不是运行时间太长或内存消耗太快,就是优化层度不高,只有文[9]有较好的运行效果,使用 CNP 量子门库可综合最长为 12 的任一长度最优的电路,但还只能同时综合前 4 层的全部长度最优的电路.文[10]提出可任意初始化量子寄存器的电路;文[11]实现对量子操作的有效分解,从而可优化综合算法.

收稿日期:2007-11-26;修回日期:2008-06-18
基金项目:国家自然科学基金(No. 60572071);国家自然科学基金重大研究计划(No. 90412014);江苏省自然科学基金(No. BK2007104, BK2008209);江苏省高校自然科学基金(No. 06KJB520137)

本文在我们前期研究成果即基于哈希函数的3量子电路快速综合算法的基础上,巧妙构造最短置换编码,对量子电路进行特定的拓扑变换,无损压缩 n 量子最优电路的存储空间近 $2\times n!$ 倍,通过已生成的最优电路的双向级联,可使用多种量子门,采用最小长度标准,以极高效率生成较长的4量子最优电路,并对国际标准的4量子测试电路进行了全面优化,在联想深腾1800主节点上,仅历时11h,率先生成了基于CNT量子门库的前8层全部共3120218828个4量子最优电路,压缩最优电路的存储空间达47.95倍,在此基础上,可快速综合长度不超过16的任意最优电路,而这些都是其他算法无法做到的.

2 量子可逆逻辑电路综合的基本概念

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,此电路必须是可逆的,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 $TOF(C,T)$,其中输入变量集合 $In=\{x_0,x_1,\cdots,x_{n-1}\}$,控制端集合 $C=\{x_{i_2},x_{i_3},\cdots,x_{i_n}\}$,受控端集合 $T=\{x_{i_1}\}$,且 $C\cap T=\varnothing$,$C\cup T\subset In$. 输出变量集合映射为 $\{x_0,x_1,\cdots,x_{i_1-1},x_{i_1}\oplus\prod_{k=2}^{n}x_{i_k},x_{i_1+1},\cdots,x_{n-1}\}$. 若 $\exists m\in\{2,3,\cdots,n\}$,$x_{i_m}=0\Rightarrow\prod_{k=2}^{n}x_{i_k}=0$,受控端 x_{i_1} 的输出为 $x_{i_1}\oplus\prod_{k=2}^{n}x_{i_k}=x_{i_1}\oplus0=x_{i_1}$;若 $\forall m\in\{2,3,\cdots,n\}$,$x_{i_m}=1\Rightarrow\prod_{k=2}^{n}x_{i_k}=1$,受控端 x_{i_1} 的输出为 $x_{i_1}\oplus\prod_{k=2}^{n}x_{i_k}=x_{i_1}\oplus1=\bar{x}_{i_1}$. 如图1所示,当 $n=1$ 时,$C=\varnothing$,$TOF(x_{i_1})$,为非门,简称 N;当 $n=2$ 时,$C=\{x_{i_2}\}$,$TOF\{x_{i_2},x_{i_1}\}$ 为控制非门,简称 C;当 $k=3$ 时,$C=\{x_{i_1},x_{i_2}\}$,$TOF\{x_{i_2},x_{i_3},x_{i_1}\}$ 为标准Toffoli门,简称 T;当 $k>3$ 时,$C=\{x_{i_2},x_{i_3},\cdots,x_{i_k}\}$,$TOF(x_{i_2}x_{i_3}\cdots x_{i_k},x_{i_1})$ 为通用Toffoli门,简称GT;其中非门为单量子比特门,其他均为多量子比特门.

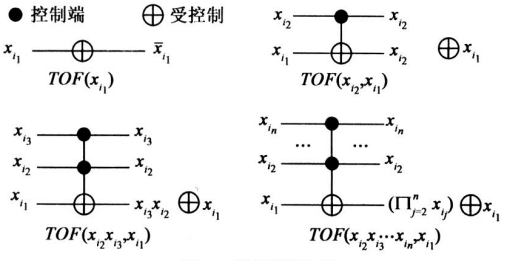


图1 量子逻辑门

定义2 Peres门,记为 $P(x_2,x_1,x_0)$,简称 P,其中输入变量集合 $In=\{x_0,x_1,x_2\}$,输出变量集合映射为 $\{x_1x_2\oplus x_0,\ x_2\oplus x_1,\ x_2\}$,其功能相当于量子门 $TOF(x_1x_2,x_0)$ 与 $TOF(x_2,x_1)$ 级联.

定义3 Peres门的逆门,记为 $PI(x_0,x_1,x_2)$,简称PI,其中输入变量集合 $In=\{x_0,x_1,x_2\}$,输出变量集合映射为 $\{x_2\oplus x_1x_2\oplus x_0,x_2\oplus x_1,x_2\}$,其功能相当于量子门 $TOF(x_2,x_1)$ 与 $TOF(x_1x_2,x_0)$ 级联.P门、PI门与大多数量子门不同,其他量子门在输入与输出对换后,功能不变,而这两个门却发生了变化,因此P门与PI门是两种不同的量子门,若文献[8]等文献中考虑同时使用这两个门,其综合结果会变得更为优化.

表1 真值表

Input		Output	
$\langle x_2,x_1,x_0\rangle_2$	$x_2x_1x_0$	$\langle y_2,y_1,y_0\rangle_2$	$y_2y_1y_0$
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4 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 m 个数,即 $|M|=m$,可称 σ 为 m 元置换.置换通常表示为 $\sigma=\begin{pmatrix}1&2&\cdots&m\\p_1&p_2&\cdots&p_m\end{pmatrix}$,置换群中都是从1开始.为方便计算,本文算法对置换进行了功能扩充,置换的数据从0开始,同样可实现各种置换运算,设 $\sigma=\begin{pmatrix}0&1&\cdots&m-1\\p_0&p_1&\cdots&p_{m-1}\end{pmatrix}$,可记为 $(p_0,p_1,\cdots,p_{m-1})$,m 元置换也可表示为不交的轮换之积.设轮换 $\tau=(a_1a_2a_3\cdots a_i)$,$i\leqslant m$,则 τ 的映射关系是 $a_1\mapsto a_2\mapsto a_3\cdots\mapsto a_i\mapsto a_1$. 而 σ 逆置换可表示为 $\sigma^{-1}=\begin{pmatrix}p_0&p_1&\cdots&p_{m-1}\\0&1&\cdots&m-1\end{pmatrix}$,$M$ 上的恒等函数 $\pi_e=\begin{pmatrix}0&1&\cdots&m-1\\0&1&\cdots&m-1\end{pmatrix}$.可逆函数可用如表1的真值表描述,其中:$\langle x_{n-1},x_{n-2},\cdots,x_0\rangle_2=\sum_{i=0}^{n-1}(x_i\cdot2^i)$,$\langle y_{n-1},y_{n-2},\cdots,y_0\rangle_2=\sum_{i=0}^{n-1}(y_i\cdot2^i)$.也可用整数集合 $\{0,1,\cdots,2^n-1\}$ 的置换表示.图2是一个3量子可逆逻辑电路,用真值表描述见表1,用置换表示为 $\sigma=\begin{pmatrix}0&1&2&3&4&5&6&7\\2&6&0&1&7&3&5&4\end{pmatrix}=(2,6,0,1,7,3,5,4)=(0\ 2)(1\ 6\ 5\ 3)(4\ 7)$,设其中NOT门,Toffoli门,CNOT门的置换分别为 $\sigma_1,\sigma_2,\sigma_3$,得 $\sigma=\sigma_1\circ\sigma_2\circ\sigma_3$.显然,量子电路的置换等价于这些量子门的置换的乘积.

为避免在本文中重复说明,特作如下说明:

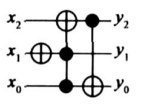


图2 量子可逆逻辑电路

(1)电路 C 表示长度为 l 的 n 量子可逆逻辑电路,由量子门 $g_1, g_2, \cdots, g_l$ 级联而成,记为 $C = g_1 g_2 \cdots g_l$.

(2)常用的三个集合定义为:

$SNF = \{1,2,\cdots,n!\}$, $SM = \{1,2,\cdots,m\}$, $SB = \{0,1\}$.

(3)已知置换 $\sigma = (0,1,\cdots,n-1)$共有 $n!$ 种不同置换,设分别为 $\sigma_1, \sigma_2, \cdots, \sigma_{n!}$.

(4)本文所指电路都是量子可逆逻辑电路,是一种特殊的量子电路,但两者的综合算法相似.

(5)$\pi(g)$表示量子门 g 的置换.电路 C 的置换为 $\pi(C) \equiv (g_1 g_2 \cdots g_l) = \pi(g_1)\pi(g_2)\cdots\pi(g_l)$.

定义 5 能用可逆函数描述的电路称为可逆逻辑电路.如图 2 所示.该电路特点是:(1)输入线数与输出线数相等;(2)没有扇出与扇入;(3)没有反馈;(4)电路分层级联,有时为保证电路可逆需要人为添加一些辅助位,如表 1 所示的真值表,设最多有 m 个相同输出,则至少要增加$\lceil \log_2^m \rceil$位辅助位,确保电路可逆.

定义 6 量子门库 $L_{n,gates}$表示在 n 条量子线上,用基本量子门 $gates$ 构成全部量子门的集合,即长度为 1 的全部最优电路,简写为 L, $T(L)$表示使用库 L 中的量子门综合的所有 $n \times n$ 量子可逆逻辑电路集合[8]

定义 7 $minl(a)$表示 $T(L)$中任意量子电路 a 的最小长度,即该电路至少由 $minl(a)$个库 L 中的量子门级联而成. $maxl(T(L))$表示 $T(L)$中所有量子电路的最小长度的最大值,简写为 $maxl(T)$[8].

由于算法中频繁使用置换,如何表示成便于形式化演算和节省空间的形式是算法效率的关键之一,为此我们提出了最短编码方案.即将置换抽象成顺序数序列(中介数),再将此序列映射成一个整数,该整数就是编码.所谓最短编码是指该整数占用的比特数最小,从而节省存储空间.

定义 8 在定义 4 的置换 $P = (p_0, p_1, \cdots, p_i, \cdots, p_{2^n-1})$中,$p_i$ 的顺序数记为 O_p^i,是指排在 p_i 前面且比小 p_i 的元素个数,可表示为

$$O_p^i = \sum_{j=0}^{i-1} \mathrm{sgn}(p_i - p_j), \mathrm{sgn}(x) = \begin{cases} 1, x > 0 \\ 0, x \leqslant 0 \end{cases} \quad (1)$$

由各个元素的顺序数组成的序列称为顺序数序列,如$(O_p^0, O_p^1, \cdots, O_p^{2^n-1})$,其中 $O_p^0 \equiv 0$, $O_p^{2^n-1} \equiv p_{2^n-1}$,如定义 4 中置换 σ 的顺序数序列为(0,1,0,1,4,3,4,4).

引理 1 在定义 8 中,p_i 的顺序数也为

$$O_p^i = p_i - \sum_{j=i+1}^{2^n-1} \mathrm{sgn}(p_i - p_j) \quad (2)$$

证明 由式 1 可知,置换 P 中排在 p_i 前面且比 p_i 小的元素个数为 O_p^i,排在 p_i 后面且比 p_i 小的元素个数为$\sum_{j=i+1}^{2^n-1} \mathrm{sgn}(p_i - p_j)$,两数之和是 P 中比 p_i 小的元素个数.已知 P 是由 $0,1,\cdots,2^n-1$ 共 2^n 个整数组成,比 p_i 小的元素为 $0,1,\cdots,p_i-1$,共有 p_i 个,所以 $O_p^i + \sum_{j=i+1}^{2^n-1} \mathrm{sgn}(p_i - p_j) = p_i$,移项后命题得证.

定理 1 计算定义 8 中置换的顺序数序列仅需比较元素 $2^{2n-2} - 2^{n-1}$次.

证明 计算顺序数序列的常用方法是只使用式 1 或式 2 计算,其比较元素次数均为 $num = \sum_{i=1}^{2^n-1} i = \sum_{i=1}^{2^n-1}(2^n - 1 - i) = 2^{2n-1} - 2^{n-1}$.为减少元素的比较次数,可根据条件动态选择这两个计算公式.即 $O_p^i = \begin{cases} \text{公式 1 计算} & i \leqslant 2^{n-1}-1 \\ \text{公式 2 计算} & i > 2^{n-1}-1-i \end{cases}$,结果显然有$\sum_{i=0}^{2^{n-1}-1} i + \sum_{i=2^{n-1}}^{2^n-1}(2^n - 1 - i) = 2^{2n-2} - 2^{n-1}$,其值小于 num 的一半.

定义 9 置换的编码或置换值的计算公式为

$$Code(p_0, p_1, \cdots, p_{2^n-1}) = \sum_{i=1}^{2^n-1} (O_p^i \cdot i!)$$

引理 2 在定义 8 中,若两个置换对应的顺序数序列相同,则这两个置换必定相同.

证明 借鉴归纳法,设任取置换 $P = (p_0, p_1, \cdots, p_{2^n-2}, p_{2^n-1})$, $Q = (q_0, q_1, \cdots, q_{2^n-2}, q_{2^n-1})$,若它们对应相同的顺序数序列 $B = (b_0, b_1, \cdots, b_{2^n-2}, b_{2^n-1})$,则 $p_i = q_i, i \in \{0,1,\cdots,2^n-1\}$,若可证得 $p_{2^n-1-i} = q_{2^n-1-i}, i \in \{0,1,\cdots,2^n-1\}$,结果成立,即 $P = Q$.设 S_P 表示置换 P 中全部元任意素构成的集合. $Min(i,S)$表示任意集合 S 中第 i 小的元素,满足$|\{d \in S \mid d \leqslant Min(i,S)\}| = i$.令 $S_0 = \{0,1,\cdots,2^n-1\}$, $S_p = \{p_{2^n-j}, p_{2^n-j+1}, \cdots, p_{2^n-1}\}$, $S_q = \{q_{2^n-j}, q_{2^n-j+1}, \cdots, q_{2^n-1}\}$.

(1)归纳基础:当 $i = 0$ 时,已知 $O_p^{2^n-1} = b_{2^n-1}$, $O_q^{2^n-1} = b_{2^n-1}$而根据式 1 可知,$p_{2^n-1} = Min(b_{2^n-1}+1, S_P)$,又因为 $S_P = S_0$,因此 $p_{2^n-1} = Min(b_{2^n-1}+1, S_0)$.同理 $q_{2^n-1} = Min(b_{2^n-1}+1, S_0)$,则 $p_{2^n-1} = q_{2^n-1} = b_{2^n-1}$,即 $p_{2^n-1-i} = q_{2^n-1-i}$.

(2)归纳假设:当 $i \in \{0,1,\cdots,j-1\}$时,$p_{2^n-1-i} = q_{2^n-1-i}$.

(3)归纳步骤:当 $i = j$ 时,已知 $O_p^{2^n-1-j} = b_{2^n-1-j}$,在置换 P 中,$p_{2^n-1-j} = Min(b_{2^n-1-j}+1, S_0 - S_p)$,同理 $q_{2^n-1-j} = Min(q_{2^n-1-j}+1, S_0 - S_q)$,由假设可知,$S_p = S_q$,因此 $S_0 - S_p = S_0 - S_q$,所以 $p_{2^n-1-j} = q_{2^n-1-j}$,即 $p_{2^n-1-i} = q_{2^n-1-i}$.

引理 3 函数 $Code$ 将不同顺序数序列生成不同编

码.

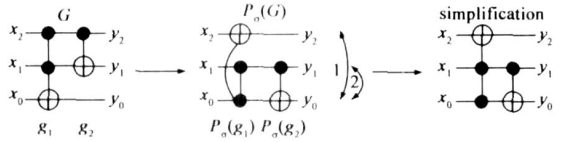
证明 已知 $c = Code(p_0, p_1, \cdots, p_{2^n-1}) = \sum_{i=1}^{2^n-1} O_p^i \cdot i!$，即该函数可将顺序数序列 $B = (O_p^0, O_p^1, \cdots, O_p^{2^n-1})$ 生成编码 c，c 除以 2 的余数为 O_p^1，商为 $\sum_{i=2}^{2^n-1} O_p^i \cdot \frac{i!}{2}$，此商除以 3 的余数为 O_p^2，依次类推. 递推公式为 $n_1 = c$，$n_{i+1} = \left\lfloor \frac{n_i}{i+1} \right\rfloor$，$O_p^i = n_i - (i+1)n_{i+1}$，$i \in \{1, 2, \cdots, 2^n - 1\}$. 显然此式可对进行逆计算，生成顺序数序列 B，因此编码与顺序数序列是一一对应，即不存在不同顺序数序列对应相同编码，即不同顺序数序列对应的编码一定不相同.

定理 2 函数 $Code$ 是生成不同置换的唯一最短编码.

证明 已知置换 $\sigma = (p_0, p_1, \cdots, p_{2^n-1})$ 共有 $2^n!$ 个不同置换，且每个置换出现概率相同，因此每个置换至少要用 $\lceil \log_2^{2^n!} \rceil$ bit 表示，而 $\min(\sum_{i=1}^{2^n-1} O_p^i \cdot i!) = \sum_{i=1}^{2^n-1} 0 \cdot i! = 0$，$\max(\sum_{i=1}^{2^n-1} O_p^i \cdot i!) = \sum_{i=1}^{2^n-1} \max(O_p^i) i! = \sum_{i=1}^{2^n-1} i \cdot i! = 2^n! - 1$，根据定义 9 得，此编码的位数满足最短编码的要求. 下面只要证明不同置换对应不同编码，即具有唯一性. 根据引理 2 知，不存在不同置换对应相同的顺序数序列，即不同置换对应的顺序数序列一定不相同，再根据引理 3 可得，不同的顺序数序列生成的编码不相同，因此，不同的置换生成编码也不相同.

定义 10 拓扑变换是指几何图形在连续变换中满足条件(a)不断裂，(b)不粘合，(c)两个点不合成一个点，变换前后两图形上的点一一对应. 拓扑性质是指几何图形在拓扑变换中保持不变的那些性质. 本文利用对最优量子电路进行特定的拓扑变换，实现最优电路的无损压缩，节省内存，扩大可综合电路的规模.

定义 11 线置换是指量子电路中各量子门与量子线的交点不变，量子线间的顺序发生变化，即量子线发生置换，显然它属于拓扑变换，设量子门 g 经过线置换 σ 后，生成的新量子门记为 $P_\sigma(g)$. 设电路 G 中有两个量子门 g_1, g_2，经过线置换 σ 后，电路变为 $P_\sigma(G) = P_\sigma(g_1 g_2) = P_\sigma(g_1) P_\sigma(g_2)$，令 $\sigma = \begin{pmatrix} 0 & 1 & 2 \\ 2 & 0 & 1 \end{pmatrix} = (0\ 2)(0\ 1)$ 得



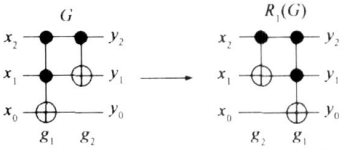
其中，双向箭头线表示两条线对换.

量子门中同类型端点间顺序的变化不影响其功能，由此可对拓扑变换后的电路进行化简.

$P_\sigma(C) = P_\sigma(g_1 g_2 \cdots g_l) = P_\sigma(g_1) P_\sigma(g_2) \cdots P_\sigma(g_l)$. 由定义 6 可知，设量子门库 L 中共有 m 个基本的量子门，分别为 $g_1, g_2, \cdots, g_m$，显然有 $L = \bigcup_{j \in SNF, i \in SM_j} P_\sigma(g_i)$，$L$ 中所有量子门的置换的集合为 $\sigma = \bigcup_{j \in SNF, i \in SM_j} P_\sigma(\pi(g_i))$.

设任意电路集合 S，$P_\sigma(S) = \{P_\sigma(G) \mid G \in S\}$ 为集合 S 中每个电路进行线置换 σ.

定义 12 量子电路的向变换有正向、逆向两种变换，正向变换则没有任何变化，而逆向变换是将电路的输入端与输出端交换，对应于真值表的输入列与输出列对换，因此新电路的置换是原电路置换的逆. 设量子门 g 的向变换记为 $R_b(g)$，当 $b = 0$ 时为正向，$R_0(g) = g$，当 $b = 1$ 时为逆向，$R_1(g) = g^{-1}$. 若 $g = g^{-1}$，则 g 为对称门，否则为非对称门，目前常用量子门中只有 peres 门是非对称门. 设原电路中有两个量子门，且是对称门，$G = g_1 g_2$，则经过逆向变换后，电路变为 $R_1(G) = R_1(g_1 g_2) = g_2^{-1} g_1^{-1} = g_2 g_1$，得



$R_1(C) = R_1(g_1 g_2 \cdots g_l) = (g_1 g_2 \cdots g_l)^{-1} = g_l^{-1} g_{l-1}^{-1} \cdots g_2^{-1} g_1^{-1}$. 若电路中均为对称门，则 $R_1(C) = g_l g_{l-1} \cdots g_2 g_1$，为原电路逆序排列. 在综合算法中，只有量子门是对称的或者它的逆门存在于量子门库中，才能使用向变换.

设任意电路集合 S，$R_b(S) = \{R_b(G) \mid G \in S\}$ 为 S 中每个电路进行向变换 b.

由定义 11 与定义 12，显然有以下结论.

性质 1 $P_\sigma(P_\tau(C)) = P_{\tau \circ \sigma}(C)$，同样 $P_\sigma(P_{\sigma^{-1}}(C)) = P_{\sigma^{-1}}(P_\sigma(C)) = C$，但 $P_\sigma(P_\tau(C)) \neq P_\tau(P_\sigma(C))$. 其中 $P_\sigma(P_\tau(C))$ 表示对电路 C 先执行线置换 τ，再执行线置换 σ 后得到的电路，显然它满足置换的乘法，即该操作等价于 $P_{\tau \circ \sigma}(C)$. 已知 $\sigma \circ \sigma^{-1} = \sigma^{-1} \circ \sigma = \pi_e$，可得 $P_{\sigma^{-1}}(P_\sigma(C)) = P_\sigma(P_{\sigma^{-1}}(C)) = P_{\pi_e}(C) = C$；已知 $\tau \circ \sigma \neq \sigma \circ \tau$，可得 $P_\sigma(P_\tau(C)) \neq P_\tau(P_\sigma(C))$.

性质 2 $R_{b1}(R_{b2}(C)) = R_{b2}(R_{b1}(C)) = P_{b1 \oplus b2}(C)$，其中 $R_{b1}(R_{b2}(C))$ 表示对电路 C 先执行向变换 $b1$，再执行向变换 $b2$ 后得到的电路. 当 $b1 = b2$，电路无向变换；当 $b1 \neq b2$，电路有向变换，显然 $R_{b2}(R_{b2}(C)) = R_{b1 \oplus b2}(C)$，其他同理.

性质 3 $R_b(P_\sigma(C)) = P_\sigma(R_b(C))$. $R_b(P_\sigma(C))$ 表示先对电路 C 执行线置换 σ，再执行向变换 b 后得到的

电路,$P_\sigma(R_b(C))$表示先对电路 C 执行向变换 B,再执行线置 σ 换后得到的电路,根据电路的特点,线置换是纵向变换,而向变换是横向变换,这两个操作的功能与操作的顺序无关,因此 $R_b(P_\sigma(C)) = P_\sigma(R_b(C))$.

据性质 1～3 可知,任一电路的拓扑变换过程中,除线置换之间是顺序有关,其他变换均为顺序无关.

引理 4　任意量子电路经过任意线置换与向变换后,其量子代价不变,即 $cost(C) = cost(R_b(P_\sigma(C)))$. 其中 $cost(C)$ 表示量子电路 C 的量子代价,$cost(g)$ 表示量子门 g 的量子代价.

证明　根据电路特点,任意量子门 g 经过任意线置换 σ 后,在电路中的位置发生变化,但类型没变,因此该门的代价不变;向变换只是电路方向发生变换,因此电路的代价不变,电路 C 拓扑变换后,$cost(R_b(P_\sigma(C))) = cost(P_\sigma(C)) = cost(P_\sigma(g_1)\ P_\sigma(g_2)\ \cdots P_\sigma(g_l)) = \sum_{i=1}^{l} cost(P_\sigma(g_l)) = \sum_{i=1}^{l} cost(g_l) = cost(C)$

已知最小代价标准是 C 的功能不变的前提下要求 $cost(C)$ 最小,若 $\forall i \in \{1,2,\cdots,l\}$, $cost(g_i) = 1$,则 $cost(C) = l$,即要求电路的长度最小,则最小代价标准退化为最小长度标准.

引理 5　任意最优电路,经过任意线置换和向变换后,生成的电路一定是最优的.

证明　用反证法证明,设最优电路 C,令电路 $D = R_b(P_\sigma(C))$,假设电路 D 不是最优,则一定存在最优电路 E,满足 $\pi(E) = \pi(D)$ 且 $cost(E) < cost(D)$,则 $\pi(R_b(P_{\sigma^{-1}}(E))) = \pi(R_b(P_{\sigma^{-1}}(D))) = \pi(R_b(P_{\sigma^{-1}}(R_b(P_\sigma(C))))) = \pi(C)$,设电路 $F = R_b(P_{\sigma^{-1}}(E))$,由引理 4 可知,$\pi(F) = \pi(C)$,又因为,$cost(F) = cost(E)$,$cost(D) = cost(C)$,根据假设 $cost(E) < cost(D)$ 得 $cost(F) < cost(C)$,所以电路 F 比相同功能的电路 C 优化,与电路 C 为最优矛盾.

引理 6　根据定义 11 可证,$\forall k \in SNF$, $P_{\sigma_k}(L) = L$.

证明　根据置换原理知,$\forall i, j \in SNF$, $\sigma_i\sigma_j \in \{\sigma_1, \sigma_2, \cdots, \sigma_{n!}\}$,若 $i \neq j \rightarrow \sigma_i \neq \sigma_j \rightarrow \sigma_i\sigma_k \neq \sigma_j\sigma_k$,因此 $\forall k \in SNF$,$\{\sigma_1\sigma_k, \sigma_2\sigma_k, \cdots, \sigma_{n!}\sigma_k\}$ 中没有相同元素,得

$$\{\sigma_1\sigma_k, \sigma_2\sigma_k, \cdots, \sigma_{n!}\sigma_k\} = \{\sigma_1, \sigma_2, \cdots, \sigma_{n!}\} \overset{同理}{=} \{\sigma_k\sigma_1, \sigma_k\sigma_2, \cdots, \sigma_k, \sigma_{n!}\} \quad (3)$$

得 $\forall k \in SNF$, $L_{new} = P_{\sigma_k}(L) = P_{\sigma_k}(\bigcup_{j\in SNF, i\in SM} P_{\sigma_j(g_i)}) = \bigcup_{\substack{i\in SNF\\ i\in SM}} P_{\sigma\sigma_k}(g_i) \overset{式3}{=} \bigcup_{i\in SM}(\bigcup_{j\in SNF} P_{\sigma_j}(g_i)) = \bigcup_{\substack{j\in SNF\\ i\in SM}} P_{\sigma_j}(g_i) = L$

定义 13　最优量子电路 G 的最小置换电路定义为:电路 G 经过全部线置换与向变换后生成的新电路中其置换值最小的电路,简称为电路 G 的最小置换电路,用 $Min(G)$ 表示,其中置换值计算公式参见定义 9. 据引理 5 可知,若 G 为最优的,则 $Min(G)$ 也一定最优,因此本文中所指的最小置换电路一定是最优电路. 量子电路 G 经过全部线置换与向变换生成电路构成的集合为 $S = \bigcup_{j\in SNF, b\in SB} R_b(P_{\sigma_j}(G))$,则 $Min(\pi(G)) = \min\{R_b(P_{\sigma_j}(\pi(G))) \mid j \in SNF, b \in SB\}$, $\exists k \in SNF$, $\exists c \in SB$, $R_c(P_{\sigma_k}(\tau)) = \min(\pi(G))$,则 $Min(G) = R_c(P_{\sigma_k}(G))$. 设 $GetMin(\tau, \sigma, b)$ 函数的是根据电路置换 τ,经过线置换 σ 向变换 b 后得到最小置换 π,即 $\pi = R_b(P_\sigma(\tau)) = Min(\tau)$,函数返回 π 值.

定理 3　电路 $Min(G)$ 是定义 13 中集合 S 的代表,它可完整无损地表示 S,即将集合 S 压缩近 $2 \times n!$ 倍.

证明　显然集合 S 中的任何电路的最小置换电路都是 $Min(G)$,则只要证明将此电路进行全部的线置换后生成的全部电路构成的集合必然与 S 相等即可. 因为 $S = \bigcup_{j\in SNF, b\in SB} R_b(P_{\sigma_j}(G))$, $Min(G) = R_c(P_{\sigma_k}(G))$,则将 $Min(G)$ 进行全部线置换生成的全部电路构成的集合为 S',则可得:

$$S' = \bigcup_{j\in SNF, b\in SB} R_b(P_{\sigma_j}(Min(G))) = \bigcup_{j\in SNF, b\in SB} R_b(P_{\sigma_j}(R_c(P_{\sigma_k}(G)))) \overset{性质3}{=} \bigcup_{j\in SNF, b\in SB} R_{b\odot c}(P_{\sigma\sigma_k\sigma_j}(G)) \overset{式3}{=} \bigcup_{j\in SNF, b\in SB} R_b(P_{\sigma_j}(G)) = S$$

因此由最小置换电路 $Min(G)$ 可无损还原最优电路集合 S,已知上述拓扑变换共有 $2 \times n!$ 种,若每种变换生成的电路都不相同,则 $|S| = 2 \times n!$,在很少情况下变换后的电路存在少量重复,因此 $|S| \approx 2 \times n!$,所以对集合 S 的压缩率近似为 $2 \times n!$ 倍.

若电路 $G, D \in S$ 且 $\exists h, j \in SNF$, $b, c \in SB$, $R_b(P_{\sigma_h}(G)) = R_c(P_{\sigma_j}(D))$ 则 $Min(D) = Min(G)$.

定义 14　设电路集合为 S 与 Q,定义 $S \times Q = \{G_1G_2 \mid G_1 \in S \wedge G_2 \in Q\}$ 为 S 与 Q 中的电路级联.

引理 7　任意电路集合 S 与 Q,一定满足等式:$P_\sigma(S \times Q) = P_\sigma(S) \times P_\sigma(Q)$.

证明　根据定义 14 得 $S \times Q = \{G_1G_2 \mid G_1 \in S \wedge G_2 \in Q\}$,则

$$P_\sigma(S \times Q) = P_\sigma(\{G_1G_2 \mid G_1 \in S \wedge G_2 \in Q\}) = \{P_\sigma(G_1G_2) \mid G_1 \in S \wedge G_2 \in Q\} = \{P_\sigma(G_1)\ P_\sigma(G_2) \mid P_\sigma(G_1) \in P_\sigma(S) \wedge P_\sigma(G_2) \in P_\sigma(Q)\} \overset{定义14}{=} P_\sigma(S) \times P_\sigma(Q)$$

定理 4　设长度为 l 的全部最优电路集合为 S_l,则 $Min(S_{l+1}) = Min((\bigcup_{b\in SB} R_b(Min(S_l))) \times L) = Min(S_l \times L)$

证明　设从 S_l 中任取电路 G,根据定义 13 知,$\exists k \in SNF$, $\exists c \in SB$, $R_c(P_{\sigma_k}(G)) = Min(G)$,显然 k、c 只与 G 有关,可令 $k(G) = k$, $c(G) = c$, $R_{c(G)}(P_{\sigma_{k(G)}}(G)) = Min(G)$ 则

$$\bigcup_{b\in SB} R_b(Min(G)) = \bigcup_{b\in SB} R_{b\odot c(G)}(P_{\sigma_{k(G)}}(G))$$

$$= \bigcup_{b\in SB} R_b(P_{\sigma_k(G)}(G)) \quad (4)$$

又因为 S 是长度为 l 的最优电路的全部,任意量子电路 $G\in S_l$, G 一定是最优的,根据引理 5 得, $\forall k\in SNF$, $\forall c\in SB$, $R_c(R_{\sigma_k}(G))$ 一定也是最优的,且长度也为 l,因此必有 $R_c(P_{\sigma_k}(G))\in S_l$,因此

$\bigcup_{\substack{j\in SNF\\ C\in SB,\ G\in S_l}} R_c(P_{\sigma_j}(G)) \subseteq S_l$,又有 $\bigcup_{\substack{j\in SNF\\ C\in SB,\ G\in S_l}} R_c(P_{\sigma_j}(G)) \supseteq \bigcup_{G\in S_l} R_0(P_{\pi_e}(G)) = S_l$,所以

$$\bigcup_{\substack{j\in SNF\\ C\in SB,\ G\in S_l}} R_c(P_{\sigma_j}(G)) = S_l \quad (5)$$

令 $c=0$, $\bigcup_{\substack{j\in SNF\\ G\in S_l}} P_{\sigma_j}(G) \subseteq S_l$, $\bigcup_{\substack{j\in SNF\\ G\in S_l}} P_{\sigma_j}(G) \supseteq \bigcup_{G\in S_l} P_{\pi_e}(G) = S_l$,

得

$$\bigcup_{j\in SNF,\ G\in S_l} P_{\sigma_j}(G) = \bigcup_{j\in SNF} P_{\sigma_j}(S_l) = S_l \quad (6)$$

则 $\bigcup_{j\in SNF} P_{\sigma_j}((\bigcup_{b\in SB} R_b(M_s)) \times L) \xrightarrow{\text{引理7}} \bigcup_{j\in SNF} P_{\sigma_j}(\bigcup_{b\in SB} R_b(M_s)) \times P_{\sigma_j}(L) \xrightarrow{\text{引理6}} \bigcup_{\substack{j\in SNF\\ G\in S_l}} P_{\sigma_j}(\bigcup_{b\in SB} R_b(Min(G))) \times L \xrightarrow{\text{式(4)}} \bigcup_{\substack{j\in SNF\\ b\in SB\\ G\in S_l}} P_{\sigma_j}(R_b(P_{\sigma_k(G)}(G))) \times L = \bigcup_{\substack{j\in SNF\\ b\in SB\\ G\in S_l}} R_b(P_{\sigma_{k(G)}\circ\sigma_j}(G)) \times L = \bigcup_{G\in S_l}(\bigcup_{\substack{l\in SNF\\ b\in SB}} R_b(P_{\sigma_k(G)}\circ\sigma_j(G)) \times L \xrightarrow{\text{式(3)}} \bigcup_{\substack{j\in SNF,\\ b\in SB,\\ G\in S_l}} R_b(P_{\sigma_j}(G)) \times L \xrightarrow{\text{式(5)}} S_l \times L$. 又因为 $\bigcup_{j\in SNF} P_{\sigma_j}(S_l \times L) \xrightarrow{\text{引理7}} \bigcup_{j\in SNF} P_{\sigma_j}(S_l) \times P_{\sigma_j}(L) \xrightarrow{\text{式(6)}} = S_l \times L$. 可得 $\bigcup_{j\in SNF} P_{\sigma_j}((\bigcup_{b\in SB} R_b(M_s)) \times L) = \bigcup_{j\in SNF} P_{\sigma_j}(S_l \times L) \xrightarrow{\text{定理3}} \Rightarrow \min((\bigcup_{b\in SB} R_b(M_s)) \times L) = Min(S_l \times L) = Min(S_{l+1})$.

为节省内存,我们只保存每个最小置换电路 $Min(S_l)$,计算 $Min(S_{l+1})$ 的常用方法是,先将 $Min(S_l)$ 解压为 S_l,然后与量子门库 L 中的每个门级联,再对生成的长度为 $l+1$ 的全部最优电路 S_{l+1} 进行压缩生成 $Min(S_{l+1})$,而本文方法只需直接将 $Min(S_l)$、$R_1(Min(S))$ 分别与量子门库 L 中的每个门级联,再对生成的长度为 $l+1$ 的部分最优电路进行压缩,同样也生成 $Min(S_{l+1})$. 显然通常方法与本文方法相比增加解压 $2\times n!\times|Min(S_l)|$ 个电路的置换,且电路的级联数量与电路置换压缩的数量匀提高近 $n!$ 倍.

3 量子可逆逻辑电路综合的新算法

每个量子逻辑门的本质是实现数据的某种置换,量子电路是若干量子门的级联,因此量子电路功能本质是若干数据置换的叠加,即置换的乘积. 我们前期提出基于 Hash 函数的 3 量子电路的快速综合算法,其思想是采用广度优先构建最优电路,将 Hash 表中的每个不同的元素与不同功能的电路建立一一对应的关系,则只需一步就能判断任意电路是否有相同功能的电路存在,快速判断电路是否最优,但如果直接将该算法用在 4 量子电路中是不可行的,因为 Hash 表长度至少为. 设二进制数中每位表示一电路,则 $n^2|_{n=4}$ = 20922789888000, b = 2435.73GB,内存占用太大. 如果使用深度优先搜索的综合算法,虽然内存占用较小,但运行速度太慢或无法达到最优或较优而失去综合的意义,因此我们选择运行速度较快,优化层度较高的广度优先搜索的综合算法,但此算法内存消耗很大,为此本算法使用 3 种节省内存的无损数据压缩方法:(1) 将表示电路功能的置换用最短的编码表示,采用定义 9 的置换编码方法表示 4 量子电路的置换仅需 45b,而通常的编码方法是选择置换中 16 个数的前面或后面的 15 个数,分别用 4b 表示,则每个置换需 15 ×4 = 60b. (2) 采用前文提出的拓扑变换,对最优电路进行无损压缩,则使用线置换与向变换压缩近 $n!\times 2|_{n=4}$ = 48 倍,仅使用线置换压缩近 $n!|_{n=4}$ = 24 倍,这是本文算法成功的关键. (3) 算法的数据结构中顶层采用长度为 2^{16} 的 Hash 表,而 Hash 表中的每个元素都指向下层的一棵不同的红黑树(RB-Tree). 顶层采用 Hash 表不仅提高数据查询速度,而且它指向的红黑树中每个节点的置换编码都减少了 2 个字节,使用红黑树是为了在内存动态按需分配的基础上保持较好的数据查询速度.

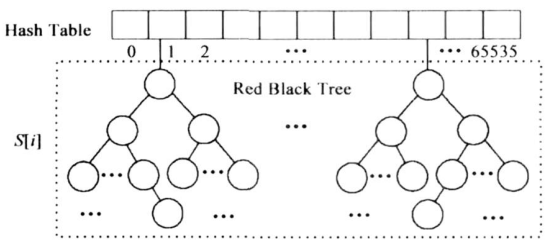


图3 第i层的最小置换电路的数据结构

如何确定 Hash 表的长度呢? 设当前 n 量子电路的置换为π,置换的最小编码为 $Code(Min(\pi))$,其二进制数形如 $\{0,1\}^{\lceil\log_2 n!\rceil}$,截取此数后面 $k\times 8$ 位,作为 Hash 地址,余下的 $\lceil\log_2 n!\rceil - k\times 8$ 位数作为树中的节点值. 设 Hash 表中各元素为 jB,则因有 Hash 表而多占用 $j\times 2^{8k}$B,而树中每个节点少占用 kB,则共少占用了 $k\times|S[i]|$B,因此本结构最大化节省内存函数为 $\max fm(k,j,i) = k\times|S[i]|1 - j\times 2^{8k}$,以基于 CNT 量子门库的 4 量子电路为例,Hash 表中每个元素是指向树的指针,占 4B,已知长度为 8 的最小置换电路的数量为 58777916,当 $k=2$,可得 $fm(2,4,8)$ = 117293688 为函数的最大值,此时 Hash 表长为 $2^{8k}|_{k=2}$ = 65536.

本文算法先广度优先搜索生成前 N 层的全部 4 量子长度最优的电路,若使用的量子门确定,则这些最优电路也就确定,因此最小置换电路也是确定的,为节省计算时间,算法将最多可计算的前 N 层全部最小置换电路保存成一个文件,若使用 CNT 量子门,则 $N=8$,文件有 700MB,因只保存最小置换电路,保存的电路数量

减少 47.95 倍,整体综合最优电路的长度从原来的 6 层增加到 8 层,整体综合最优电路数量增加 117.7 倍,可综合的电路长度从 12 增加到 16.

3.1　最小长度整体综合算法

设红黑树中节点的数据类型为:

```
struct rbtnode
{gate;//本电路最后的量子门.
cpm;//本电路的最小置换.
binv;//本电路变成最小置换电路时是否反向.
lnpm;//本电路变成最小置换电路时所用线置换.
pcpm;//指向上一层电路的置换.
pbinv;} //前面电路是否反向
```

为增强算法的可读性,作如下两点简化. (1) 置换没有用最小置换编码表示,(2) 略去顶层的 Hash 表,每层最小置换电路只保存在一棵红黑树中,如图 3 中 $S[i]$就是保存全部长度 i 的最小置换电路的一棵树.

设量子电路有 n 条量子线,量子门库中 m 有个不同量子门,即 m 个不同置换规则,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. 根据广度优先方法从 0 出发依次生成全部 1 个门、2 个门等等,直到个门的最小置换电路. 从仅需 0 个量子门的恒等电路开始,集合 $S[0]$中仅有一个节点,即$(\mathrm{cpm}:\pi_e)$. 显然0 个量子门一定是最小置换电路,设已有所有长度为 $l-1$ 的最小置换电路,则所有长度为 l 的最小置换电路的生成方法参见定理4,判断集合 $S[0..l]$中是否存在 π,如果没有则写入 $S[l]$,否则说明此前一定存在功能相同且不长于当前电路的电路,这是因为算法中电路是按长度从小到大依次生成,因此要去除,依次类推,直至生成全部量子电路的置换都存在或内存不足为止.

算法1　最小长度量子电路整体综合算法 *QML*.

输入:量子门库 L.

输出: $maxl$, $N[0..maxl]$, $S[0..maxl]$.

1. $S[0]=\{(cpm:\pi_e)\}$, $j=0$, $N[0]=1$
2. while $N[j]\neq 0$ do{
3. $j=j+1$, $S[j]=\varnothing$
4. for all $Nodex$ in $S[j-1]$do{
5. $c=Nodex$, cpm
6. for $v=0$to 1do{
7. if $v=1$then $c=c^{-1}$
8. for all a in L do{
9. $p=c\cdot a$, $\pi=GetMin(p,\sigma,b)$
10. if $\pi\notin\cup_{i=0}^{j}S[i]$then{
11. $S[j]=S[j]\cup\{(gate:g_a, cpm:\pi,$ $lnpm:\sigma, binv:b, pcpm:c, pbinv:v)\}$
12. if memory overflow error then
13. $\{free\ S[i], j=j-1,$goto15$\}\}\}\}$
14. $N[j]=\sum_{G\in S[j]}\left|\bigcup_{i\in SNF,\,b\in SB}R_b(P_{\sigma_i}(G))\right|\}$
15. $maxl=j$

算法 *QML* 同时生成尽可能多的最小长度量子电路,统计出各长度的电路总数,并返回其最大长度. 第 1 步 $S[0]$中只包含一个恒等电路,实现恒等置换 π_e, $j=0$ 表示电路没有门, $N[j]=1$ 表示长度为 0 的电路总数为 1;第 2 步表示若还存在长度为 $j+1$ 的最优电路,则试求长度为 $j+1$ 的最优电路;第 3 步长度 j 增加 1,最优电路的置换构成的集合置为空;第 4 步依次取出长度为 $j-1$ 的最优电路对应节点;第 5 步 c 置为节点的置换;第 6～7 步对前 $j-1$ 层电路进行两种向变换,第 8～12 步在长度为 $j-1$ 的最优电路 c 后面追加量子门库中的量子门,即将置换 c 依次与量子门 L 库中的每个量子门对应的置 a 换乘积,生成新的置换 p;其中, $GetMin(p,\sigma,b)$ 功能见定义 13;第 10 步判断在已有集合中是否存在置换π,若不存在;第 11 步将量子门 g_a,即在量子门库中置换为 a 的量子门、当前电路的最小置换π、线置换的逆 σ^{-1}、向变换 b、前层电路的最小置换 c、前层电路向变换 v 构成的节点插入到 $S[j]$中;第 12～13 步判断如果内存溢出,将释放当前层的 $S[j]$,提前结束;第 14 步统计 $S[j]$表示的全部最优量子电路的个数,存入中 $N[j]$;第 15 步返回可整体综合电路的最大长度.

3.2　具体量子电路序列生成算法

因为综合算法中增加了拓扑变换,导致综合过程非常复杂. 设所求电路存在于已生成的长度为 l 的最小置换电路,则生成该电路序列算法如下.

算法2　算法 *QML* 构造的最小置换电路的电路序列生成算法 *QMC*.

输入:量子门库 L,最小置换电路的置换为 P,电路长度为 l.

输出:所求量子电路序列.

1. do $QML(L)$,生成最小置换电路集合 $S[0..maxl]$.
2. $i=l$, $mynode[i]=S[i].find(p)$
3. $pcpmx=mynode[i].pcpm$
4. while $i>1$do{
5. $i=i-1$, $mynde[i]=S[i].$find$(pcpmx)$
6. $pcpmx=mynode[i].pcpm\}$
7. $\sigma=mynode[1].lnpm$, $b=mynode[1].binv$
8. $G_1=P_\sigma R_b(mynode[1].gate)$
9. for $i=2$ to l do
10. $\{\ \sigma=mynode[i].lnpm$, $b=mynode[i].binv$, $c=mynode[i].pbinv$, $g=mynode[i].gate$
11. $G_i=R_b(P_\sigma(R_c(G_{i-1})\ g))\}$
12. retrun G_l

算法 *QMC* 是根据最小置 p 换生成电路序列. 第 1 步仅需运行一次,将 *QML*(*L*) 运算的结果保存成一个文件,以后只要打开该文件;第 2 步在已知电路长度为的情况下,直接在 $S[l]$ 中查询电路置换为 p 的节点. 第 3 步得到当前节点对应量子门的前面相连量子门的置换;第 4～6 步是不断向前找出电路中全部节点;第 7～8 步根据第一个量子门的拓扑变换信息推出前面长度为 1 的最小置换电路;第 9～11 步不断从前向后依次进行拓扑变换,分别生成前面长度为 2,3, …, l 的最小置换电路. 其中第 11 步 $R_c(G_{i-1})\ g$ 表示先对电路 G_{i-1} 按 c 进行向变换,然后将得到的电路与量子门 g 级联;第 12 步返回所求电路序列.

算法 3: 具体量子电路序列生成算法 *QMR*. 输入:量子门库 L,所求量子电路的置换 p.

输出:所求长度最小的量子电路序列.

1. if $\exists\, l \in \{0,1,2,\cdots,maxl\}$, $GetMin(p,\sigma,b) \in S[l]$ then
2. { $G = QMC(L, GetMing(p,\sigma,b), l)$
3. return $R_b(P_{\sigma^{-1}}(G))$ }
   else if $\exists\, i \in SNF$, $\exists\, b \in SB$, $\exists\, l\{1,2,\cdots,maxl\}$,
4. $\exists\, Nodex \in S[l],)\ GetMin(R_b(P_{\sigma_i}(p)))\circ (Nodex.cpm)^{-1}, \tau, c) \in S[maxl]$ then
5. { $G_2 = QMC(Nodex, cpm, l)$,
6. $\rho = GetMin(R_b P_{\sigma_i}(p))\circ (Nodex.cpm)^{-1}, \tau, c)$
7. $G_1 = QMC(\rho, maxl)$
8. return $R_b(P_{\sigma_i^{-1}}(R_c(P_{\tau^{-1}}(G_1)))(G_2))$ }
9. else{ return *NULL*}

算法 *QMR* 是综合一个具体的 4 量子可逆逻辑电路. 第 1 步是判断所求电路的最小置换是否存于已生成的电路中,如果存在,则在第 2～3 步中生成具体的电路序列,否则在第 4 步中判断是否存在所求电路经过某种拓扑变换后的置换为 $R_b(P_{\sigma_i}(p))$,此置换作为长度为 $l \in \{1,2,\cdots,maxl\}$,输入置换为 π,置换为 $Nodex.cpm$ 的电路输出,则 $\pi\circ(Nodex.cpm) = R_b(P_{\sigma_i}(p)) \Rightarrow \pi = R_b(P_{\sigma_i}(p))\circ(Nodex, cpm)^{-1}$,若 π 的最小置换存在于 $S[maxl]$,则通过第 5～8 步生成电路序列,否则说明此电路的长度一定超过 $2\times maxl$ 而无法综合.

4 实验结果与分析

在综合长度最优的 4 量子电路的实验中,因算法复杂度很高,只有文[9]涉及此问题. 文[9]基于 CNP 量子门库,以最小长度为标准,在文[8]综合全部 4 层电路的基础上,通过双向综合增加 4 层,结合深度优先搜索又增加 4 层,从而实现可综合最多为 12 层的任意电路,但文[9]只能同时综合全部前 4 层最优电路. 本文算法使用 CNT 量子门库,以最小长度为标准,可生成全部 8 层最优电路,采用特定双向级联,在不增加内存消耗的情况下,可快速生成最多长度为 16 的最优的量子电路,本文中的双向级联与文[9]的双向综合有较大不同,文[9]是从电路的首与尾两处分别计算,向中间综合,而本文是先综合前面电路,再直接将其移到后面经特定拓扑变换后使用,避免重复计算.

为验证本文算法综合复杂电路的能力,我们还采用 Maslov 提供的 4 量子标准测试电路进行实验. 以基于 CNT 量子门的两个电路 4_49、Hwb4 为例,通过综合发现这两个电路都是长度最优的,将这两个电路级联,见图 4 的电路 A,前面 12 个门为电路 4_49,后面 11 个门为电路 Hwb4,在调用 *QML* 生成电路整体综合的数据文件的基础上运行,仅历时 35S 就生成了电路序列 B,可以验证,电路 A、B 的置换都为(15,2,3,12,5,9,1,11,0,10,14,6,4,8,7,13). 本算法还随机综合了大量电路,还没有发现无法综合的电路.

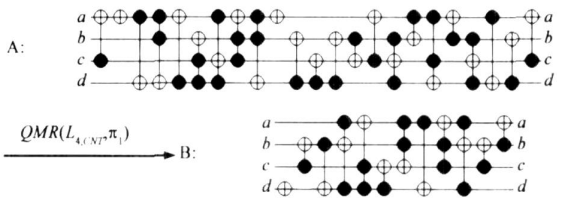


图4 量子可逆逻辑电路优化

5 结论

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,提出一种新颖高效的量子电路综合算法,巧妙构造置换的最短编码,利用量子电路特定的拓扑变换,压缩量子最优电路的存储空间近倍,通过已生成最优电路的双向级联,可使用多种量子门,采用最小长度量子代价标准,以极高效率生成最优的 4 量子可逆逻辑电路,下面我们将该程序升级为并行程序,在深腾 1800 集群系统中进行并行计算,按照统计数据推测,有望实现快速综合任意 4 量子最优可逆逻辑电路,从而解决这一公认的难题.

致谢 美国波特兰州立大学宋晓宇教授对本课题提出许多宝贵建议,在此表示衷心感谢.

参考文献:

[1] D Deutsch. Quantum theory, the Church-Turing principle and the universal quantum computer [J]. Proc Royal Soc London, 1985, 400(1818): 97 - 117.

[2] E Fredkin, T Toffoli. Conservative logic [J]. International Journal of Theoretical Physics, 1982, 21: 219 - 253.

[3] X Y Song, G W Yang, M Perkowski, et al. Algebraic characteristics of reversible gates [J]. Theory of Computing Systems,

2005,39(2):311 - 319.

[4] D Maslov, G W Dueck, D M Miller. Toffoli network synthesis with templates [J]. IEEE Trans on Circuits and Systems-I, 2005,24(6):807 - 817.

[5] W Q Li, H W Chen, Z Q Li. Application of semi-template in reversible logic circuit [A]. Proceedings of the 11th International Conference on CSCWD [C]. Melbourne, Australia, 2007. 155 - 161.

[6] P Gupta, A Agrawal, N KJha. An algorithm for synthesis of reversible logic circuits [J]. IEEE Trans on Circuits and Systems-I, 2006,25(11):807 - 817.

[7] V V Shende, A K Prasad, I L Markov, et al. Synthesis of reversible logic circuits [J]. IEEE Trans on Circuits and Systems-I, 2003,22(6):723 - 729.

[8] G W Yang, X Y Song, M Perkowski, et al. Fast synthesis of exact minimal reversible circuits using group theory [A]. Proceedings of IEEE ASP-DAC 2005[C]. Shanghai, China, 2005. V2,18 - 21.

[9] G W Yang, X Y Song, W N N Hung, M Perkowski. Bi-directional synthesis of 4-bit reversible circuits [J]. The Computer Journal, 2008,51(2):207 - 215.

[10] G L Long, Y Sun. Efficient scheme for initializing a quantum register with an arbitrary superposed state [J]. Phys Rev A, 2001,64(1):014303:1 - 8.

[11] J J Vartiainen, M Mottonen, M M Salomaa. Efficient decomposition of quantum gates [J]. Phys Rev Lett 2004,92(17):177902:1 - 4.

作者简介:

李志强　男,1974 年生于江苏姜堰,博士研究生,讲师. 主要研究方向为量子计算、可逆电路综合与测试. E-mail:yzqqLzq@163.com

陈汉武　男,1958 年生于江苏南京,博士,教授,博士生导师,主要研究方向为量子计算、信息论.

徐宝文　男,1961 年生江苏东台,博士,教授,博士生导师,研究领域为程序设计语言,软件智能化.

肖芳英　女,1982 年生安徽省,博士研究生,主要研究方向为量子计算、量子电路测试与量子编码.

薛希玲　女,1985 年生山东省,硕士研究生,主要研究方向为量子算法与仿真.