

基于可逆函数复杂性的正反控制门可逆网络综合

倪丽惠¹ 管致锦² 聂志浪²

(南通大学电子信息学院 南通 226019)¹ (南通大学计算机科学与技术学院 南通 226019)²

摘要 提出了基于可逆函数复杂性的正反控制(PNC, Positive/Negative Control)门可逆网络综合方法。该方法根据可逆函数的输出排列,逐次交换输出向量,在交换过程中减少函数的复杂性,直至复杂性为零。每一次向量的交换对应一个独立的PNC门。利用该方法综合部分可逆函数,并将实验结果与国际上代表性文献采用的实验用例进行比较。结果表明,构造的可逆网络在可逆门数上有一定程度的改善。

关键词 可逆网络, PNC门, 复杂性

中图法分类号 TP301.6 **文献标识码** A

Reversible Network Synthesis for Positive/Negative Control Gates Based on Reversible Function's Complexity

NI Li-hui¹ GUAN Zhi-jin² NIE Zhi-lang²

(College of Electronics and Information, Nantong University, Nantong 226019, China)¹

(College of Computer Science and Technology, Nantong University, Nantong 226019, China)²

Abstract A methodology on synthesizing reversible network for positive/negative control (PNC) gates based on reversible function's complexity was proposed in this paper. According to the reversible function's output permutation, we exchanged the positions of two output vectors step by step to reduce the function's complexity until the complexity was zero. Each output switching corresponded to an individual PNC gate. Through synthesizing part of reversible functions and comparing the results with the international representative cases used in other literatures, our reversible network created by the proposed method had an improvement in the number of gates.

Keywords Reversible network, PNC gates, Complexity

1 引言

可逆逻辑综合是可逆计算的重要内容^[1,2],被广泛地应用于低功耗 CMOS 电路设计^[3]、量子计算及光计算中。按照 Landauer 理论,传统逻辑门中信息位的丢失会引起能量的损耗。每比特信息的丢失会产生 $kT \times \ln 2$ 焦耳的能耗,其中 k 是波尔茨曼常量, T 是绝对温度。比如:传统的或门是二输入一输出的,存在一个信息位的丢失。

Bennett 证明了由可逆门级联的电路在理论上能实现零能耗^[4]。可逆门是指输入与输出之间存在一一对应关系的逻辑门,亦即由输入可以得到输出,由输出也可以唯一地确定输入。常用的可逆逻辑门有 CNOT 门、CCNOT 门和 Fredkin 门^[5],这些逻辑门的输入和输出之间不存在信息位的丢失。

可逆门综合的主要任务是按照可逆网络无扇出、无反馈等约束条件和限制,根据给定的可逆函数的输出排列,通过级联一系列可逆门来实现对应的可逆电路。已有不少学者提出他们各自的方法来实现综合。V. Shende 等人基于枚举法,给出了相应的网络构造方法^[6]。Robert Wille 提出了基于 BDD

(Binary Decision Diagram, 二叉决策图)实现可逆网络的综合方法^[7],根据该方法得到的可逆网络的规模由 BDD 控制。D. M. Miller 利用双向置换方法,提出了以 Toffoli 门为基础的可逆网络级联算法,并给出了相应的网络化简方法^[8]。Mishchenko 和 Perkowski 提出了可逆波级联的规则结构^[9],并且证明了实现这样的结构需要的级联函数比 ESOP 积项要少。但这个方法只适合于小规模可逆逻辑综合,而且不容易测量,同时它还需要一个可逆的规范。

本文提出了基于正反控制(PNC, Positive/Negative Control)门的可逆综合方法,该方法的核心思想是根据给定的可逆函数的输出排列,逐次交换输出向量,减少函数的复杂性,直至复杂性为零。对于相同的可逆函数,采用不同的综合方法,得到的结果是不相同的。本文对一些具有代表性的可逆函数进行综合,并与已有文献中的结果相比较,结果表明,本文构造的可逆级联网络在平均门数方面有一定的改善。

2 基本定义

PNC 门是在一般 Toffoli 门的基础上进行衍变和推广而

到稿日期:2009-11-30 返修日期:2010-02-10 本文受国家自然科学基金(60873069),国家高技术研究发展计划(863 计划)项目(2007AA01Z404)和南通大学博士基金(09B07)资助。

倪丽惠(1986—),女,硕士生,主要研究方向为可逆计算、信息安全等, E-mail: nilihuiandpzl@126.com;管致锦(1962—),男,博士,教授,主要研究方向为可逆计算、逻辑综合、信息安全等, E-mail: guan_zj@ntu.edu.cn(通信作者);聂志浪(1986—),男,本科生。

得到的。基于 PNC 门的可逆级联网络的基本元素有两类控制线、一类目标线和一条无关线^[10]。

定义 1 基于 PNC 门的可逆级联网络的基本元素由下面 5 种线型构成^[10]，如见图 1 所示。

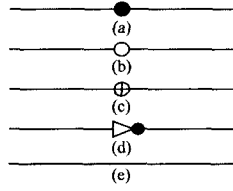


图 1 正反控制模型的 5 种基本线型

1) 肯定控制线(见图 1(a))。如果这条线上的输入是 0，则目标线的值将保持不变。如果输入是 1，则目标线的值是否发生改变将由其它的肯定/否定控制线上的输入情况来确定。通过肯定控制线的值不变。

2) 否定控制线(见图 1(b))。如果这条线上的输入是 1，则目标线的值将保持不变。如果输入是 0，则目标线的值是否发生改变将由其它的肯定/否定控制线上的输入情况来确定。通过否定控制线的值不变。

3) 目标线(见图 1(c))。每一个 PNC 门中有且只有一个目标线。通过目标线的值受到肯定/否定线的控制。

4) 否定线(见图 1(d))。通过否定线的值取反。

5) 无关线(见图 1(e))。通过这条线的值不发生任何变化。

定义 2 用带有方向且标明权向量的图来表示一个 n 输入/输出的可逆函数 F ，称这样的图为转向图，记为 $G_F(V, E)$ 。其中集合 $V = [0, 1, \dots, 2^n - 1]$ ，表示 2^n 个输入向量。设 $F(i) = \mu_i, i = 0, 1, \dots, 2^n - 1, U = \{\mu_i\}$ 。集合 $E = \{e_{jk} | F(k) = j, j \in U, k \in V\}$ ，表示 2^n 个输出向量 j 到输入向量 k 的对应关系。输出排列、转向图及集合 E 这三者是可以互推的。

比如：易知图 4(a) 对应函数的有向集合 $E = \{e_{00}, e_{11}, e_{22}, e_{44}, e_{37}, e_{75}, e_{53}, e_{66}\}$ ，且对应函数的输出排列为 $F = [0, 1, 2, 5, 4, 7, 6, 3]$ 。其中 E 的子集 e_{ll} (比如 e_{11} 及 e_{22}) 表示元素 l 自成循环。

输出向量 j 和输入向量 k 之间存在权向量，记为：

$$W_{jk} = [\omega_{jk}(n), \omega_{jk}(n-1), \dots, \omega_{jk}(1)] \quad (1)$$

式中， $\omega_{jk}(m) = j(m) \oplus k(m), 1 \leq m \leq n, j(m)$ 和 $k(m)$ 分别是向量 j 和 k 的第 m 个变量值。比如：在图 4(a) 中， $W_{43} = [1, 1, 1]$ 及 $W_{37} = [1, 0, 0]$ 等。

每一个权向量 W_{jk} 都有对应的绝对权，记为：

$$|W_{jk}| = \sum_{m=1}^n \omega_{jk}(m) \quad (2)$$

绝对权 $|W_{jk}|$ 也就是向量 j 和 k 之间的汉明距离。

对于一个 n 输入/输出的可逆函数 F ，可用复杂性 $C(F)$ 作为衡量其性能的一种方式，记为：

$$C(F) = \sum_{e_{jk} \in E} |W_{jk}| \quad (3)$$

即集合 E 中所有 e_{jk} 的绝对权 $|W_{jk}|$ 之和。比如：图 4(a) 所对应函数的复杂性为 4。

给定一个 n 输入/输出的可逆函数 F ，如果两个输出向量之间的汉明距离为 1 (即只有第 m 个变量的值不同)，则交换这两个输出向量，称以上交换为“输出交换”(简称 OS, Output Switching)。每一次 OS 操作都对应一个 PNC 门，多次 OS 操作对应多个 PNC 门的级联。

本文将能够减少函数复杂性的 OS 记为“D-OS”(Decreased Output Switching)，即 $C(F_i) < C(F)$ 。类似地，保持函数复杂性不变的 OS 记为“E-OS”(Equivalent Output Switching)，即 $C(F_i) = C(F)$ 。增加函数复杂性的 OS 记为“I-OS”(Increased Output Switching)，即 $C(F_i) > C(F)$ 。

在文中，将存在“D-OS”的函数称为“可减函数”。但是，有些可逆函数本身或者在经过若干次输出向量的交换后，不存在“D-OS”，称这样的函数为“不可减函数”。对于一个 n 输入/输出的不可减函数 F ，如果转换图中存在 3 个连续的向量 i, j 和 k ，且 $|W_{ij}| = 1$ ，则对向量组 $\{i, j\}$ 执行 OS 操作的前提是交换前后函数的复杂性保持不变。互换向量后，在原先的转换图中，向量 j 会形成自循环(e_{jj})，而向量 i 和 k 会成为连续的向量(e_{ik})，且伴有 $W_{ik} = W_{ij} + W_{jk}$ ，上述 E-OS 操作记为“合 E-OS”。

类似地，如果转换图中存在两个连续的向量 i 和 j ，且 $|W_{ij}| > 1$ ，则选择一个向量 l 且对向量组 $\{i, l\}$ 执行 OS 操作的前提是 l 和 i 只在第 m 个变量上不同，且交换前后函数的复杂性保持不变。互换向量后，生成了 3 个连续的向量 i, l 和 j (e_{il} 和 e_{lj})，上述 E-OS 操作记为“分 E-OS”。

定理 1 对于一个 n 输入/输出的不可减函数 F ，如果转换图中存在 $k+2$ 个连续的向量 $v_0, v_1, \dots, v_{k+1}$ ，且它们满足以下 3 个条件：

$$(1) |W_{v_n v_{n+1}}| = 1, \forall n \in [0, 1, \dots, k-1]$$

$$(2) W_{v_0 v_1} \neq W_{v_1 v_2} \neq \dots \neq W_{v_{k-1} v_k}$$

$$(3) w_{v_0 v_1}(m) = 1, \text{ 且 } w_{v_k v_{k+1}}(m) = 1$$

则存在 $k-1$ 个向量组： $\{v_k, v_{k-1}\}, \dots, \{v_2, v_1\}$ ，依次对这些向量组执行 OS 操作后，能将原先的“不可减函数”转换为“可减函数”，并将这 $k-1$ 个向量组记为“T-OS”(Transitional Output Switching)。

证明：依次交换 $k-1$ 个向量组 $\{v_k, v_{k-1}\}, \dots, \{v_2, v_1\}$ 后，向量 $v_2, v_3, \dots, v_k$ 都已自成循环， v_0, v_1, v_{k+1} 这 3 个向量重新组成连续的向量，且存在 $W_{v_1 v_{k+1}} = \sum_{i=1}^k W_{v_i v_{i+1}}$ 。因为 $w_{v_0 v_1}(m) = 1$ 且 $w_{v_1 v_{k+1}}(m) = w_{v_k v_{k+1}}(m) = 1$ ，所以在新的函数中肯定存在一组“D-OS”，即 $\{v_0, v_1\}$ 。因此，依次交换 $k-1$ 个向量组后，原先的“不可减函数”将转换为“可减函数”。

3 正反控制门的可逆综合

可逆逻辑综合，是指用给定的可逆门，根据可逆函数的输出排列，在满足无扇出及反馈的前提下，通过一系列门的级联构成相应的可逆网络。本文只利用 PNC 门实现可逆函数的综合。

Zheng Yexin 等人提出了以 Toffoli 门为基础的综合方法^[11]。该方法从给定的可逆函数开始，在输出端添加可逆门，通过添加门来减少函数的复杂性。有时可能找不到能够减少复杂性的可逆门，但在任何时候都能够找出不增加函数复杂性的可逆门作为过渡门，使得函数的复杂性能再次减少。然而，该方法在处理含有极端情况的函数时，频繁使用过渡门，从而大大增加了级联网络的实现代价。

本文引入 PNC 门，当找不到能够减少函数复杂性的可逆门时，每添加一个过渡门就会执行一次查找，以确保添加的过渡门数最少。综合考虑，提出了基于 PNC 门的可逆综合算法。

3.1 PNC 门的生成与级联

根据第 2 节的一系列概念及定理,提出了针对 PNC 门的可逆综合算法,该算法通过在输出端依次添加 PNC 门的方式来实现可逆网络。按照可逆函数与“D-OS”的关系,该算法存在以下 3 种情况,具体执行情况如图 2 所示。

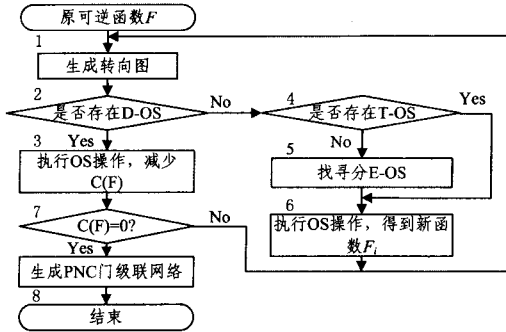


图 2 基于 PNC 门的综合算法流程图

情况 1 可逆函数中一直存在“D-OS”,对于这种情况,重复步骤 1、步骤 2、步骤 3 和步骤 7 即可得到最终的 PNC 门级联网络。

情况 2 原函数一开始或执行数次“D-OS”操作后,新的函数中不存在“D-OS”,且此时函数的复杂性不为零,则需要找寻“T-OS”,即执行步骤 4,目的是使得新生成的函数再次成为“可减函数”。

情况 3 可逆函数中既不存在“D-OS”,也找不到“T-OS”,则此时应找寻“分 E-OS”,即执行步骤 5,每执行一次“分 E-OS”都要重新确定新函数 F_i 中是否存在“D-OS”,即重复执行步骤 1。经过有限次“分 E-OS”操作,总能使得函数中重新存在“D-OS”。

执行一次 OS,对应一个 PNC 门。将所有用到的 PNC 门逆序级联,即可得到给定函数 F 的可逆级联网络。

3.2 实例验证

这部分给出 3 个可逆函数,运用本文提出的算法分别对其进行综合,来验证方法的正确性。

例 1 可逆函数 $F=[0,1,2,4,5,7,6,3]$,运用基于 PNC 门的综合算法实现该函数,其转向图如图 3(a)所示。

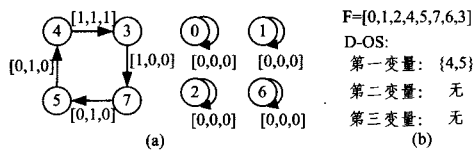


图 3 例 1 中可逆函数 F 的转向图

由转向图 3(a)可知,该函数的复杂性 $C(F)=6$ 。通过查找每一个变量,只存在一组“D-OS”,即 $\{4,5\}$,它们只在第一变量上不同,如图 3(b)所示。交换 $\{4,5\}$ 的位置后,原函数 F 的复杂性减少 2,得到了新的函数 $F_1=[0,1,2,5,4,7,6,3]$,重复步骤 1。

函数 F_1 的转向图如图 4(a)所示,其复杂性 $C(F_1)=4$ 。查找结果表明,输出向量 5 和 7 是一组“D-OS”,它们只在第二变量上不同,如图 4(b)所示。两者的位置交换后,函数 F_1 的复杂性减少 2,得到新的函数 $F_2=[0,1,2,7,4,5,6,3]$ 。在函数 F_2 中,输出向量 $(1,1,1)=F_2[(0,1,1)]$ 和 $(0,1,1)=F_2[(1,1,1)]$,易知,它们只在第三变量 $(m=3)$ 上不同,且 $\vec{w}_{73}=[1,0,0]$ 及 $\vec{w}_{37}=[1,0,0]$ 。两者互换位置后,新函数 $F_3=[0,$

$1,2,3,4,5,6,7]$ 与恒等函数完全一致,即 $C(F_3)=0$ 。

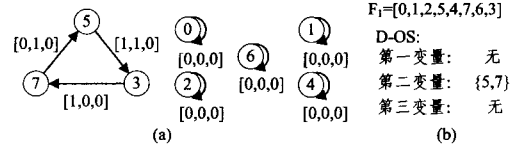


图 4 例 1 中可逆函数 F_1 的转向图

最后,将每一步交换转换成对应的 PNC 门,依次为 $PNC_3(x_3, x_2, x_1)$, $PNC_3(x_3, x_1, x_2)$ 和 $PNC_3(x_2, x_1, x_3)$ 。将上述 3 个门逆序级联,可得相应的可逆网络,如图 5 所示。

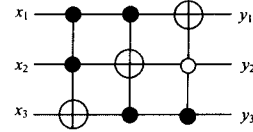


图 5 例 1 中可逆函数 F 的可逆网络

例 1 中的函数 $F=[0,1,2,4,5,7,6,3]$ 是最理想的一种综合情况,因为在函数 F 及新函数 F_i 中一直存在“D-OS”,通过有限次的输出向量互换后,得到恒等函数。但是,有些可逆函数本身或者在经过若干次输出向量的交换后,不存在“D-OS”。比如:可逆函数 $F=[4,1,2,0,5,7,6,3]$ 中不存在“D-OS”,其转向图如图 7(a)所示。下面,本文将给出综合“不可减函数”的方法。

例 2 可逆函数 $F=[4,3,2,0,5,7,6,1]$,运用基于 PNC 门的综合算法实现该函数,其转向图如图 6(a)所示。

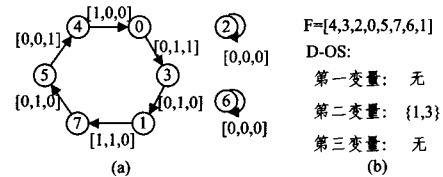


图 6 例 2 中可逆函数 F 的转向图

由图 6(b)易知,输出向量 1 和 3 是一组“D-OS”,互换位置后,得到新函数 $F_1=[4,1,2,0,5,7,6,3]$,其相应的转向图如图 7(a)所示。函数 F_1 中不存在“D-OS”,针对这种情况,转步骤 4 (见图 3) 进行处理,即根据定理 1,寻找可能存在的“T-OS”。

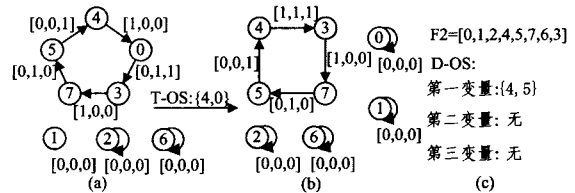


图 7 例 2 中由可逆函数 F_1 转化为 F_2 的示意图

在图 7(a)中,存在连续的输出向量 5,4,0 和 3,且它们均满足定理 1 中的 3 个条件:(1) $|W_{54}|=|W_{40}|=1$, (2) $W_{54}=[0,0,1] \neq W_{40}=[1,0,0]$, (3) $w_{54}(1)=1$, 且 $w_{03}(1)=1$, 所以存在一组“T-OS”,即 $\{0,4\}$ 。交换 $\{0,4\}$ 后,新函数 $F_2=[0,1,2,4,5,7,6,3]$ 是一个“可减函数”,其转向图及“D-OS”分别如图 7(b)和图 7(c)所示。继续分析,可得经过 3 次“D-OS”后函数的复杂性为 0。这 3 次“D-OS”对应的互换向量依次为 $\{4,5\}$, $\{5,7\}$ 和 $\{3,7\}$ 。

最后,将每一步交换转换成各自对应的 PNC 门,依次为

$PNC_3(\bar{x}_3, x_1, x_2), PNC_3(x_2, x_1, x_3), PNC_3(x_3, \bar{x}_2, x_1), PNC_3(x_3, x_1, x_2)$ 和 $PNC_3(x_2, x_1, x_3)$ 。将上述5个门逆向级联即可得到相应的可逆网络,如图8所示。

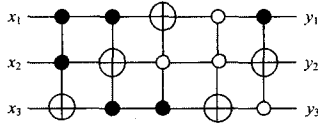


图8 例2中可逆函数F的可逆网络

在寻找“T-OS”时,结果有可能并不唯一,应选取其中最优秀的“T-OS”。其实,在转向图7(a)中,连续向量3,7,5,4和0也满足定理1中的3个条件:

- (1) $|W_{37}| = |W_{75}| = |W_{54}| = 1$
- (2) $W_{37} \neq W_{75} \neq W_{54}$
- (3) $w_{37}(3) = 1$

且 $w_{10}(1) = 1$ 。但在这种情况下,需要依次交换两组输出向量的位置,即{4,5}和{5,7},交换的组数越多,利用的PNC门越多,所以应该优先选择交换组数少的“T-OS”。

最后,考虑最极端的一种综合情况,既不存在“D-OS”也找不到“T-OS”,此时,通过找寻“分E-OS”,将“不可减函数”再次转换为“可减函数”。

例3 函数 $F = [0, 13, 2, 3, 4, 5, 6, 7, 8, 1, 9, 11, 15, 10, 14, 12]$,运用基于PNC门的综合算法实现该函数,其部分转向图如图9(a)所示。

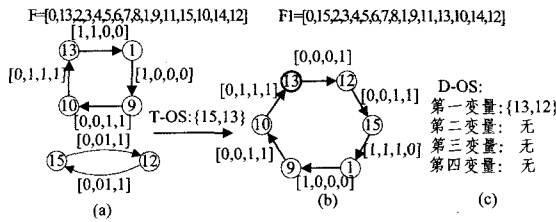


图9 例3中由可逆函数F转化为 F_1 的示意图

可逆函数 $F = [0, 13, 2, 3, 4, 5, 6, 7, 8, 1, 9, 11, 15, 10, 14, 12]$ 中,既不存在“D-OS”也找不到“T-OS”,针对这种情况,需要寻找“分E-OS”。转向图9(a)中存在两个转向环,先从向量数少的环着手。因为 $|W_{15,12}| = 2 > 1$,且 $w_{15,12}(2) = 1$,所以选取只在第二变量上与15不相同的向量,即13。向量13位于另一个环中,交换向量组{15, 13}后,函数的复杂性不发生改变,且原先的两个环合并成一个环,得到新的函数 $F_1 = [0, 15, 2, 3, 4, 5, 6, 7, 8, 1, 9, 11, 13, 10, 14, 12]$,这是一个“可减函数”,其转向图如图9(b)所示。

经过两次“D-OS”,即依次互换输出向量{13, 12}和{13, 15}后,得到一个新的“不可减函数”函数 $F_3 = [0, 13, 2, 3, 4, 5, 6, 7, 8, 1, 9, 11, 12, 10, 14, 15]$,其部分转向图如图10(a)所示。

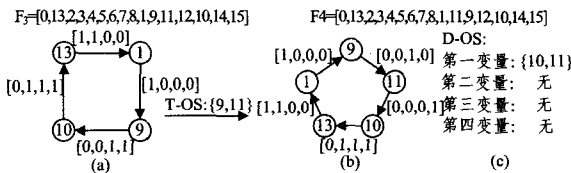


图10 例3中由可逆函数 F_3 转化为 F_4 的示意图

继续寻找“分E-OS”,并以转向图10(a)中的向量9作为查找起点。因为 $|W_{9,10}| = 2 > 1$,且 $w_{9,10}(2) = 1$,所以选取只

在第二变量上与9不相同的向量,即11。向量11已自成循环,交换向量组{9, 11}后,函数的复杂性不发生改变,且生成函数 $F_4 = [0, 13, 2, 3, 4, 5, 6, 7, 8, 1, 10, 11, 12, 9, 14, 15]$,这是一个“可减函数”,其转向图及存在的“D-OS”分别如图10(b)和图10(c)所示。

经过4次“D-OS”后,函数的复杂性为0。这4次“D-OS”对应的互换向量依次为{11, 10}, {9, 11}, {9, 13}和{9, 1}。

最后,将每一步交换转换成对应的PNC门,依次为 $PNC_4(x_4, x_3, x_1, x_2), PNC_4(x_4, x_3, \bar{x}_2, x_1), PNC_4(x_4, x_3, x_1, x_2), PNC_4(x_4, \bar{x}_3, x_1, x_2), PNC_4(x_4, \bar{x}_3, x_2, x_1), PNC_4(x_4, \bar{x}_3, x_1, x_2), PNC_4(x_4, \bar{x}_2, x_1, x_3)$ 及 $PNC_4(\bar{x}_3, \bar{x}_2, x_1, x_2)$ 。将上述8个门逆向级联即可得到相应的可逆网络,如图11所示。

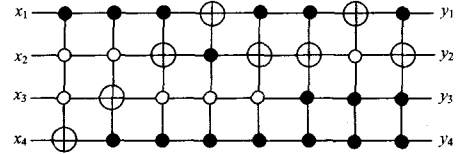


图11 例3中可逆函数F的可逆网络

3.3 可逆网络的化简

根据相应的算法构成可逆网络后,要对可逆网络进行化简,在大多数情况下,化简后的网络在门的个数、控制位数及量子代价等方面都会有一定的改善。文献[6]中引入了可逆模板,对综合后的Toffoli门级联网络进行化简。由于可利用的模板数量有限,优化的程度也受到了限制,且使用的模板越多,搜索的时间也会大大增加。

本文给出了基于PNC门的级联网络的化简方法,当网络中某些连续的PNC门满足以下情况时,即可进行化简:

- (1) 这些连续的PNC门拥有相同的目标位。
- (2) 除了目标位以外,如果不相同的控制位个数为 $n(n \geq 1)$,且这些控制位是相邻的 2^n 个组合,则可将这 n 个控制位删去,既减少了控制位,也减少了级联电路的代价。

例4 图11中的级联网络并不是最简的,可以进一步化简第5个门 $PNC_4(x_4, x_3, x_1, x_2)$ 和第6个门 $PNC_4(x_4, \bar{x}_3, x_1, x_2)$ 。这两个门的目标位相同,且控制位 x_3 互补,可将其删除,用 $PNC_3(x_4, x_1, x_2)$ 代替以上两个门,功能不变,但在门数、控制位数上得到了一定程度的改善,进而优化了可逆网络。化简后的可逆级联图如图12所示。

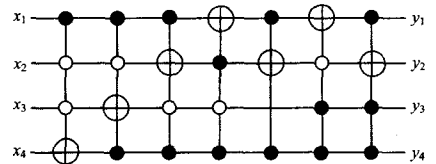


图12 例3中可逆函数F化简后的可逆网络

4 实验结果

本文提出的算法已用C++语言实现,且选取了所有3输入/输出函数及部分4输入/输出函数进行综合,以验证该算法的正确性及有效性。表1为全部3输入/输出函数的综合结果,表2为部分可逆函数对应的具体的PNC门及其级联顺序。在上述两个表中,“We”代表本文提出的综合方法,“M. Saeedi”代表文献[12]中由M. Saeedi等人提出的算法。

表1 所有3输入/输出可逆函数的综合结果及比较

门数	函数个数(We)	函数个数(M. Saeedi)
17	0	5
16	0	35
15	0	138
14	0	238
13	0	510
12	1	1362
11	89	2586
10	959	3695
9	3488	4613
8	8012	5493
7	11045	6094
6	9263	6095
5	5050	4929
4	1834	2995
3	476	1237
2	90	267
1	12	27
0	1	1
平均门数	6.775	7.440

表 2 部分可逆函数的综合结果及比较

序号	输出序列	门数		PNC 门级联顺序 (从低位到高位依次为 x_1, x_2, x_3)
		We	M. Saeedi	
1	$\{1,0,3,2,5,7,4,6\}$	4	6	$(x_3, \bar{x}_2, x_1), (x_3, \bar{x}_1, x_2), (x_2, x_1), (\bar{x}_3, \bar{x}_2, x_1)$
2	$\{7,0,1,2,3,4,5,6\}$	3	3	$(\bar{x}_2, \bar{x}_1, x_3), (\bar{x}_1, x_2), (x_1)$
3	$\{0,1,2,3,4,6,5,7\}$	3	3	$(x_3, \bar{x}_1, x_2), (x_3, \bar{x}_2, x_1), (x_3, \bar{x}_1, x_2)$
4	$\{0,1,2,4,3,5,6,7\}$	7	7	$(x_2, x_1, x_3), (\bar{x}_1, x_3, x_2), (x_3, \bar{x}_2, x_1), (\bar{x}_2, \bar{x}_1, x_3), (x_3, x_1, x_2), (x_2, x_1, x_3), (\bar{x}_2, \bar{x}_1, x_3)$
5	$\{4,3,0,2,7,5,6,1\}$	5	8	$(\bar{x}_2, \bar{x}_1, x_3), (\bar{x}_3, \bar{x}_1, x_2), (\bar{x}_3, x_2, x_1)$
6	$\{1,2,3,4,5,6,7,0\}$	3	3	$(x_2, x_1, x_3), (x_1, x_2), (x_1)$
7	$\{1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,0\}$	4	4	$(x_3, x_2, x_1, x_4), (x_2, x_1, x_3), (x_1, x_2), (x_1)$
8	$\{0,7,6,9,4,11,10,13,8,15,14,1,12,3,2,5\}$	8	3	$(x_4, \bar{x}_2, x_1, x_3), (x_4, x_1, x_2), (x_3, x_2, \bar{x}_1, x_4), (\bar{x}_4, x_2, \bar{x}_1, x_3), (\bar{x}_4, \bar{x}_2, x_1, x_3), (x_4, x_1, x_2), (\bar{x}_2, x_1, x_4), (\bar{x}_3, x_2, x_1, x_4)$
9	$\{3,6,2,5,7,1,0,4\}$	7	8	$(x_3, \bar{x}_1, x_2), (x_3, x_2, x_1), (\bar{x}_3, \bar{x}_2, x_1), (\bar{x}_2, x_1, x_3), (\bar{x}_3, x_1, x_2), (\bar{x}_2, x_1, x_3)$
10	$\{1,2,7,5,6,3,0,4\}$	7	8	$(\bar{x}_3, x_2, x_1), (x_2, \bar{x}_1, x_3), (x_3, x_2, x_1), (x_3, x_1, x_2), (x_2, x_1, x_3), (\bar{x}_3, \bar{x}_2, x_1), (\bar{x}_1, x_2)$
11	$\{0,1,2,3,4,5,6,8,7,9,10,11,12,13,14,15\}$	9	15	$(x_3, x_2, x_1, x_4), (x_4, x_2, x_1, x_3), (x_4, x_3, x_1, x_2), (x_4, x_3, \bar{x}_2, x_1), (\bar{x}_3, \bar{x}_2, \bar{x}_1, x_4), (x_4, \bar{x}_3, x_1, x_2), (x_4, x_2, x_1, x_3), (x_3, x_2, x_1, x_4), (\bar{x}_3, \bar{x}_2, x_1, x_4)$
12	$\{7,5,2,4,6,1,0,3\}$	7	6	$(x_3, x_1, x_2), (\bar{x}_3, x_2, x_1), (\bar{x}_2, x_1, x_3), (\bar{x}_3, x_2, x_1), (x_2, x_1, x_3), (\bar{x}_2, \bar{x}_1, x_3)$
13	$\{6,2,14,13,3,11,10,7,0,5,8,1,15,12,4,9\}$	16	23	$(x_4, x_3, \bar{x}_2, x_1), (x_3, x_2, \bar{x}_1, x_4), (\bar{x}_3, x_2, x_1, x_4), (x_4, x_2, x_1, x_3), (\bar{x}_4, \bar{x}_2, \bar{x}_1, x_3), (\bar{x}_4, \bar{x}_3, \bar{x}_2, x_1), (\bar{x}_4, \bar{x}_2, x_1, x_3), (\bar{x}_4, \bar{x}_3, \bar{x}_1, x_2), (\bar{x}_3, \bar{x}_2, \bar{x}_1, x_4), (x_4, x_3, x_1, x_2), (\bar{x}_3, \bar{x}_2, x_1, x_4), (\bar{x}_4, \bar{x}_2, x_1, x_3), (\bar{x}_4, x_3, \bar{x}_1, x_2), (x_4, x_3, x_1, x_2), (\bar{x}_3, x_1, x_2), (x_4, x_2, \bar{x}_1, x_3)$
平均门数		6.38	7.46	

由表 1 可知,在所有 3 输入/输出函数,即共 40320 个可逆函数中,利用本文提出的方法进行综合,最多用到 12 个 PNC 门,平均门数为 6.775;利用文献[12]中 M. Saeedi 等人提出的算法进行综合,最多需要 17 个 PNC 门,平均门数为 7.440。相比之下,本文算法在可逆门的个数方面有很大的优越性。

由表 2 可知,利用本文的综合方法得到的函数的平均门数为 6.38,比文献[12]中的平均门数 7.46 要少,大大减少了级联网络中门的个数,降低了网络成本。

结束语 本文提出了基于可逆函数复杂性的 PNC 门综合方法,并利用该方法综合了全部 3 输入/输出函数及部分 4 输入/输出可逆函数。实验结果表明,本文构造的可逆级联网络在可逆门数及控制位数方面有一定程度的改善。

在未来的工作中,我们将着手于 PNC 门的网络优化,尽可能地减少控制位的数目,降低网络的成本,使其适用于大规模可逆网络的生成。

参考文献

- [1] Schrom G, Selberherr S. Ultra-low-power CMOS Technology [C]// IEEE 9th International Semiconductor Conference. 1996; 237-246
- [2] Knill E, Laamane R, Milburn G J. A Scheme for Efficient Quantum Computation with Linear Optics[J]. Nature, 2001, 409: 46-52
- [3] De Vos A, Desoete B, Adamski A, et al. Design of Reversible Logic Circuits by Means of Control Gates[C]// Proc. Patmos 2000 Conference (Springer Lecture Notes in Computer Science No. 1918). 2000; 255-264
- [4] Bennett C H. Logical Reversibility of Computation [J]. IBM Journal of Research and Development, 1973, 17: 525-532
- [5] Maslov D, Dueck G W, Miller D M. Fredkin/Toffoli Templates for Reversible Logic Synthesis[C]// International Conference on Computer Aided Design. San Jose, USA, 2003; 256-261
- [6] Shende V V, Prasad A K, Markov I L. Synthesis of Reversible Logic Circuits[J]. IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems, 2003, 22: 710-722
- [7] Wille R, Drechsler R. BDD-based Synthesis of Reversible Logic for Large Functions[C]// Design Automation Conference. San Francisco, USA, 2009; 270-275
- [8] Miller D M, Maslov D, Dueck G W. A Transformation based Algorithm for Reversible Logic Synthesis[C]// Design Automation Conference. Anaheim, California, USA, 2003; 318-323
- [9] Mishchenko A, Perkowski M. Logic Synthesis of Reversible Wave Cascades[C]// Proceedings of International Workshop Logic and Synthesis. New Orleans, Louisiana, USA, 2002; 197-202
- [10] 管致锦, 秦小麟, 施佺, 等. 基于正反控制模型的可逆逻辑综合[J]. 计算机学报, 2008, 5: 835-844
- [11] Zheng Y-X, Huang C. A Novel Toffoli Network Synthesis Algorithm for Reversible Logic[C]// Design Automation Conference. Asia and South Pacific, 2009; 739-744
- [12] Saeedi M, Sedighi M, Zamani M S. A Novel Synthesis Algorithm for Reversible Circuits[C]// IEEE/ACM International Conference on Computer-Aided Design. San Jose, California, USA, 2007; 65-68