

基于布尔表达式图的可逆电路综合方法

卜登立, 郭 鸣

(井冈山大学电子与信息工程学院, 江西吉安 343009)

摘 要: 本文基于布尔表达式图(Boolean Expression Diagram, BED)提出一种可逆电路综合方法. 该方法使用BED表示函数, 采用逐BED结点方式综合可逆电路. 在综合一个结点时, 通过考虑其子结点函数的值是否还会被后续电路使用, 基于由NOT、CNOT以及混合极性Peres门构成的门库构建该结点的局部最优可逆子电路. 为进一步改善所得电路的成本, 根据函数表达式的乘积项中变量对的共享度对变量进行分组实现BED中变量的排序. 使用一组基准函数对所提出方法进行了验证. 结果表明所提出方法具有较高时间效率. 与现有使用决策图作为函数表示模型的综合方法相比, 所提出方法能改善综合所得可逆电路的量子成本, 且在许多情况下还能减少量子位数和垃圾线数.

关键词: 可逆电路; 逻辑综合; 布尔表达式图; 变量排序

中图分类号: TP387, TP391.72

文献标识码: A

文章编号: 0372-2112 (2020)03-0494-09

电子学报 URL: <http://www.ejournal.org.cn>

DOI: 10.3969/j.issn.0372-2112.2020.03.011

Reversible Circuit Synthesis Method Based on Boolean Expression Diagram

BU Deng-li, GUO Ming

(School of Electronics and Information Engineering, Jinggangshan University, Ji'an, Jiangxi 343009, China)

Abstract: A reversible circuit synthesis method based on BED (Boolean Expression Diagram) is proposed. The proposed method utilizes a BED to represent a Boolean function, and synthesizes a reversible circuit through mapping BED nodes to reversible cascades. Using a gate library consisting of NOT, CNOT and mixed-polarity Peres gates, the proposed method constructs a locally optimal reversible cascade for each BED node by considering whether the values of the child nodes of this node will be used by subsequent reversible cascades. In order to improve the cost of the circuit synthesized from the BED of a function, according to the sharing of pairs of variables among products of the function expression, variable ordering for the BED is achieved by variable grouping. The proposed method is validated using a set of benchmark functions. The results show that the proposed method is time efficient. Compared to the existing reversible circuit synthesis methods using decision diagram as a representation model, the proposed method can improve the quantum cost of the synthesized reversible circuits for almost all of the used benchmark functions, and can also reduce the number of qubits and the number of garbage lines in many cases.

Key words: reversible circuit; logic synthesis; Boolean expression diagram; variable ordering

1 引言

可逆逻辑的概念在20世纪60年代由Landauer引入, 并被Bennett和Toffoli进一步发展与细化^[1]. 随着半导体技术收益的递减以及对低功耗计算的日益关注, 可逆逻辑综合的研究越来越受到重视^[2]. 信息无损的

可逆计算可以避免电路因信息损失产生的能耗^[3], 这使得可逆逻辑电路被作为低功耗设计的一种替代方案^[4]. 因量子门运算的固有可逆性, 可逆逻辑成为量子计算机的基本组成形式和量子电路模型的核心部分^[5], 可逆逻辑综合也被视为量子电路综合的一个必要阶段^[6].

收稿日期: 2018-10-31; 修回日期: 2019-11-23; 责任编辑: 梅志强

基金项目: 国家自然科学基金(No. 61961023, No. 61640412, No. 61762052); 江西省教育厅科技计划项目(No. GJJ160746); 井冈山大学博士科研启动项目(No. JZB1803); 流域生态与地理环境监测国家测绘地理信息局重点实验室资助课题(No. WE2016012); 江西省自然科学基金项目(No. 20171BAB202010)

根据采用的函数表示模型,现有可逆逻辑综合方法可分为功能综合及结构综合方法^[7].文献[8]中基于轮换和文献[9]中基于变换的方法都属于功能综合方法.功能综合方法常采用真值表、正极性 Reed-Muller 展开或 Reed-Muller 谱表示函数,函数表示模型的指数级复杂度导致这类方法仅适用于规模相当小的函数^[7].使用决策图等高效数据结构能够改善功能综合方法的可扩展性^[7],如文献[10]将正极性 Reed-Muller 展开与决策图相结合,文献[11]将群论与二元决策图相结合,但以上方法综合某些输入数小于 10 的函数的时间就已达到甚至超过了 10 分钟^[10,11].此外,功能综合方法往往要求待综合函数是可逆函数^[7],对于不可逆函数,需要预先将其嵌入到可逆函数,而不可逆函数的最优嵌入问题是 coNP 难问题^[12].文献[7]中的方法将不可逆函数的嵌入结合到了可逆电路综合过程之中,但该方法在综合某些输入数小于 50 的函数时仍需若干分钟,因而不能很好地应用于大规模函数.

结构综合方法使用积之异或和(Exclusive-or Sums-Of-Products, ESOP)展开^[13]或者决策图(Decision Diagram, DD)^[14]表示函数,通过将乘积项或 DD 结点视为结构块,并将其映射为可逆子电路^[7],不仅将不可逆函数的嵌入结合到了可逆电路综合过程之中,而且能很好地处理大规模函数,并具有较高的时间效率.在综合可逆电路时,与采用 ESOP 表示模型相比,采用 DD 表示模型能够获得具有更低量子成本的结果,因此引起了人们较多的兴趣.文献[14]基于二元决策图(Binary DD, BDD)综合可逆电路,文献[15]则在此基础上,通过设计电路模板并采用子图同构技术来降低量子位数和量子成本.基于 DD 表示模型综合可逆电路时,所得电路成本与 DD 的复杂度有关,采用具有较低复杂度的 DD 表示形式或者降低 DD 的复杂度有助于降低所得电路的成本.因此,文献[16]基于正 Davio 决策图(Positive Davio DD, PDD)、文献[17]基于 Kronecker 功能决策图(Kronecker Functional DD, KFDD)综合可逆电路,文献[2]则采用了双条件 BDD(Biconditional Binary DD, BBDD)表示模型.为降低由 KFDD 综合所得可逆电路的量子成本和量子位数,文献[18]采用了在 KFDD 中使用补边并结合负控制线 Toffoli 门的方法,文献[19]则采用了对共享结点进行排序的方法.尽管这些基于 DD 表示模型的综合方法所得可逆电路具有相对较低的量子成本,但却存在着量子位数或垃圾线数较高的缺点,并且对于某些函数,量子成本仍有进一步降低的可能.从可逆逻辑的量子计算应用来看,量子位是一种非常有限的硬件资源,因此减少量子位数或垃圾线数非常重要,而高量子成本意味着高计算时间,这将会增加量子系统受噪声影响而崩溃的概率,因此降低量子

成本也是一个非常重要的设计目标^[10].

布尔表达式图(Boolean Expression Diagram, BED)^[20]是使用运算符结点对 BDD 进行扩展的一种非标准形有向无环图表示.对于一个函数,其 BED 表示有可能比其 BDD 表示具有更低的复杂度^[20],因此使用 BED 作为函数表示模型综合可逆电路有可能改善量子成本和量子位数.本文提出基于 BED 的可逆电路综合方法,该方法使用 BED 表示布尔函数,通过深度优先后序方式遍历 BED,逐结点进行可逆电路综合,并根据结点函数基于由 NOT、CNOT 和混合极性 Peres 门^[21]构成的门库构建结点的局部最优可逆子电路.为进一步降低由 BED 综合所得电路的成本,研究者提出基于变量对的共享度对变量进行分组来实现 BED 中变量的排序.本文最后使用一组 MCNC 和 RevLib 函数^[22]对所提出方法进行了验证,并与现有基于 DD 表示模型的可逆电路综合方法的结果进行了比较.

2 基本概念

2.1 可逆逻辑门库

可逆函数是输入数与输出数相等的双射,可逆门可以实现可逆函数,可逆电路则是可逆门的级联,并且不允许直接使用扇出与反馈^[1].可逆电路综合最常用的 NCT 门库包含 NOT、CNOT 和 Toffoli 门,其符号表示如图 1 所示,这 3 种门的 NCV 实现的量子成本分别为 1、1 和 5.图 1 中的“ \oplus ”表示异或运算,其所在的线为目标线,“ \bullet ”所在的线为控制线.

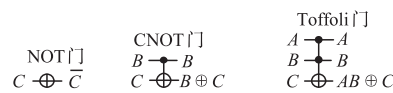


图1 NCT门库

采用新型可逆门综合可逆电路在近年来得到了较多关注,如文献[21]在综合无辅助线的可逆电路时使用混合极性 Peres(Mixed-Polarity Peres, MPP)门来改善量子成本. MPP 门包括 Peres 门(PG)、反转 Peres 门(IPG)、第一控制线为负控制线的 Peres 门(NPG)以及或型 Peres 门(OPG)^[21],图 2 给出了 MPP 门的 NCT 表示,其中 A 线和 B 线分别为第一和第二控制线, C 线为目标线. MPP 门均为 3 量子位可逆门,其 NCV 实现的量子成本都为 4.

由于可以使用 PG 和 CNOT 门构建与 Toffoli 门功能等价的电路,故本文在综合可逆电路时使用由 NOT、CNOT 和 MPP 门构成的 NCMPP 门库.

2.2 布尔表达式图

对于 n 输入变量 $\{x_i \mid 1 \leq i \leq n\}$ 的布尔函数中 $f: \{0,1\}^n \rightarrow \{0,1\}$,其积之和或 ESOP 表示如式(1)所示.

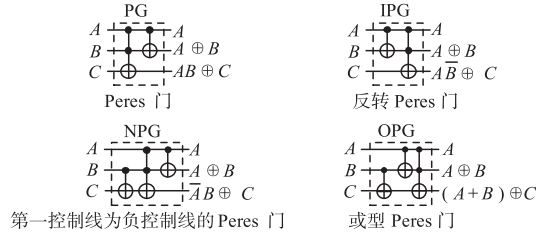


图2 混合极性Peres门的NCT表示

$$f = \bigoplus_{l=1}^t P_l \quad (1)$$

式(1)中的 $P_l = \bigwedge_{i=1}^n \hat{x}_i$ 为乘积项, $\hat{x}_i \in \{-, \bar{x}_i, x_i\}$, $\hat{x}_i = -$ 表示变量 x_i 不在乘积项 P_l 中出现, t 为乘积项的个数. 对于积之和以及 ESOP 表示, 式(1)中的“ \bigoplus ”分别为逻辑或“+”和逻辑异或“ \oplus ”.

定义 1 BED^[20] 是一个有向无环图 $G = (V, E)$, $V = \{v_j | 1 \leq j \leq N\}$ 为结点集合, E 则为结点之间边的集合. 结点 v_j 有 3 种类型: 终端结点 (包含 0 结点和 1 结点)、变量结点和运算符结点. 除终端结点外, 每个结点 v_j 都有 0 分支子结点 $v_{j,0}$ 和 1 分支子结点 $v_{j,1}$, 结点 v_j 的结点函数使用 f^{v_j} 表示.

图 3 给出了结点 v_j 的示意图. 若 v_j 是如图 3(a) 所示的 x_i 变量结点, 则 $f^{v_{j,0}} = 0, f^{v_{j,1}} = 1, f^{v_j} = x_i$. 当 v_j 是如图 3(b) 所示的运算符结点时, 假设运算符使用 op 表示, 如果 op 为 NOT, 那么 $f^{v_{j,0}} = f^{v_{j,1}} = f^{v_j}$, 否则 $f^{v_j} = f^{v_{j,0}} \text{ op } f^{v_{j,1}}$.

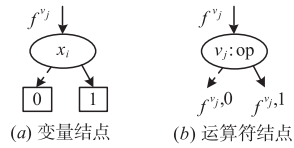


图3 BED结点示意图

定义 2 如果结点 v_k 的入度大于或等于 2, 则称 v_k 为共享结点.

定义 3 对于一个 n 输入变量的函数, 在由其函数表达式构建 BED 时, 输入变量被解析的次序称为变量顺序 π . 默认变量顺序为 $\pi_0 = \{1, 2, \dots, n\}$.

定义 4 对于函数 f_1 和 f_2 , 如果 f_2 可以通过输入取反、输入置换或输出取反变换为 f_1 , 则称 f_1 与 f_2 是 NPN 等价的^[23].

定义 5 已知结点 $w_1, w_2 \in V$, 以及结点 $u \in V \setminus \{w_1, w_2\}$; 如果任何由 u 至 0 结点或 1 结点的路径 p 都可以分解为由 u 至 $w \in \{w_1, w_2\}$ 的路径 p_1 以及由 w 至 0 结点或 1 结点的路径 p_2 , 则称 $\{w_1, w_2\}$ 为 u 的 2 运算符割^[20].

定义 6 2 运算符割化简^[20]. 如果以 u 为根结点的 BED 有一个 2 运算符割 $\{w_1, w_2\}$, 则存在一个运算符 op

可以使得 $f^u = f^{w_1} \text{ op } f^{w_2}$.

在由函数的表达式构建 BED 时, 本文除进行 NPN 等价变换以及合并同构子图外, 还进行 2 运算符割化简^[20], 以进一步减少 BED 中的结点数.

3 基于 BED 的可逆电路综合

本文在综合可逆电路时, 采用深度优先遍历的后序方式访问 BED 结点, 对于变量结点, 将其综合为一条线, 对于运算符结点, 则根据其结点函数从 NCMPP 门库中选取恰当的可逆门构建结点的局部最优可逆子电路.

3.1 BED 运算符结点的综合

运算符结点的共享有助于降低由 BED 综合所得可逆电路的量子成本, 但在综合共享结点 v_k 的父结点时, f^{v_k} 的值是否还需被后续电路使用会影响综合 v_k 的父结点所得可逆子电路的量子成本和/或量子位数, 进而影响由 BED 综合所得电路的成本. 下面先对“ f^{v_k} 的值是否还需被后续电路使用”进行解释.

假设 v_k 有 $m (m > 1)$ 个父结点, 若按深度优先、后序遍历 BED 的方式对 v_k 的父结点进行排列, 得到集合 $\{v_{k,p_j} | 1 \leq j \leq m\}$, 则在综合 $v_{k,p_j} (j < m)$ 时, f^{v_k} 的值还需被综合 $\{v_{k,p_i} | j < i \leq m\}$ 所得可逆电路使用, 仅在综合 v_{k,p_m} 时, f^{v_k} 的值才不再被后续电路使用.

如果 v_k 的入度为 1, 即 v_k 没有被共享, 仅有一个父结点 $v_{k,p}$, 那么在综合 $v_{k,p}$ 时, f^{v_k} 的值也不再被后续电路使用.

为降低由 BED 综合所得可逆电路的量子成本和量子位数, 下面以图 3(b) 所示的运算符结点 v_j 为例, 根据其子结点函数 $f^{v_{j,0}}$ 或 $f^{v_{j,1}}$ 的值是否还需被后续电路使用来构建 v_j 的局部最优可逆子电路.

当 v_j 为 NOT 结点时, $f^{v_{j,0}} = f^{v_{j,1}}, f^{v_j} = \overline{f^{v_{j,0}}}$. 如果 $f^{v_{j,0}}$ 的值还需被后续电路使用, 则需增加 1 条输入值为常量 1 的辅助线用于保存 $f^{v_j} = \overline{f^{v_{j,0}}}$ 的计算结果, v_j 的综合结果为图 4(a) 所示电路, 其量子成本 $QC(v_j) = 1$, 辅助量子位数 $AQB(v_j) = 1$; 否则 v_j 的综合结果为图 4(b) 所示电路, 此时 $QC(v_j) = 1, AQB(v_j) = 0$.

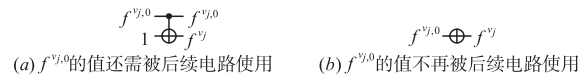


图4 NOT结点综合所得电路

由于除 NOT 外, BED 所支持的其他运算符均为双目运算符, 而对于双目运算符结点有 $f^{v_{j,0}} \neq f^{v_{j,1}}$, 因此可根据 $f^{v_{j,0}}$ 或 $f^{v_{j,1}}$ 的值是否还需被后续电路使用, 分三种情形来构建每一种双目运算符结点 v_j 的局部最优可逆子电路, 结果如表 1 所示. 其中情形 I 为“ $f^{v_{j,0}}$ 和 $f^{v_{j,1}}$ 的

值均还需被后续电路使用”,情形 II 为“ $f^{v_j,0}$ 的值不再被后续电路使用”,情形 III 则为“ $f^{v_j,1}$ 的值不再被后续电路使用”. 如果一条线的输出使用符号 G 进行标记则表示该线的输出为垃圾输出。

表 1 BED 运算符结点综合所得电路

OP	v_j 的结点函数	结点 v_j 综合所得电路			QC(v_j), AQB(v_j)		
		情形 I	情形 II	情形 III	情形 I	情形 II	情形 III
AND	$f^{v_j} = f^{v_j,0} f^{v_j,1}$				5,1	4,1	4,1
NAND	$f^{v_j} = \overline{f^{v_j,0} f^{v_j,1}}$	三种情形下 v_j 综合所得电路与 AND 类似, 只是 f^{v_j} 线的常量输入值为 1					
XOR	$f^{v_j} = f^{v_j,0} \oplus f^{v_j,1}$				2,1	1,0	1,0
BIIMP	$f^{v_j} = \overline{f^{v_j,0} \oplus f^{v_j,1}}$				2,1	2,0	2,0
OR	$f^{v_j} = f^{v_j,0} + f^{v_j,1}$				5,1	4,1	4,1
NOR	$f^{v_j} = \overline{f^{v_j,0} + f^{v_j,1}}$	三种情形下 v_j 综合所得电路与 OR 类似, 只是 f^{v_j} 线的常量输入值为 1					
IMP	$f^{v_j} = \overline{f^{v_j,0}} + f^{v_j,1} = \overline{f^{v_j,0}} \overline{f^{v_j,1}}$				5,1	4,1	4,1
NIMP	$f^{v_j} = \overline{\overline{f^{v_j,0}} + f^{v_j,1}} = \overline{f^{v_j,0}} \overline{f^{v_j,1}}$	三种情形下 v_j 综合所得电路与 IMP 类似, 只是 f^{v_j} 线的常量输入值为 0					
LIMP	$f^{v_j} = f^{v_j,0} + \overline{\overline{f^{v_j,0}} + f^{v_j,1}} = f^{v_j,0} + \overline{f^{v_j,0}} \overline{f^{v_j,1}}$				5,1	4,1	4,1
NLIMP	$f^{v_j} = \overline{\overline{f^{v_j,0}} + \overline{\overline{f^{v_j,0}} + f^{v_j,1}}} = \overline{f^{v_j,0}} + \overline{f^{v_j,0}} \overline{f^{v_j,1}}$	三种情形下 v_j 综合所得电路与 LIMP 类似, 只是 f^{v_j} 线的常量输入值为 0					

由图 4 和表 1 可以看出,当 v_j 的子结点函数的值不再被后续电路使用时,如 v_j 为 NOT、XOR 或 BIIMP 结点, v_j 的最优可逆子电路因无需增加辅助线,从而可以减少量子位数,对于 XOR 结点,还可以降低量子成本;如 v_j 为其他类型的运算符结点, v_j 的最优可逆子电路则可以降低量子成本。

3.2 基于 BED 的可逆电路综合算法

给定一个函数的积之和或 ESOP 表示,根据上述 BED 结点的综合方法,可设计如算法 1 所示的基于 BED 的可逆电路综合算法. 其中 QC 和 QB 分别表示量子成本和量子位数。

算法 1 可逆电路综合算法

Step1: QC←0, QB←0;
 Step2: 读取函数的 PLA 格式网表;
 Step3: 按照默认变量顺序 π_0 依次解析函数表达式中的每一个乘积项,构建 BED;
 Step4: 遍历 BED,对于每一个结点 v_j ,若 v_j 是变量结点则将其综合为一条线, QB←QB+1;若 v_j 是运算符结点,则按节 3.1 所述方法对其进行综合,级联可逆电路, QC←QC+QC(v_j), QB←QB+AQB(v_j)。

由算法 1 可知,由 BED 综合所得可逆电路的量子成本和量子位数与 BED 的大小有关. 而 BED 的大小与变量顺序有关系,因此由 BED 综合所得可逆电路的成

本也与变量顺序有关系。

3.3 基于变量分组的变量排序

对变量进行排序,目的是识别出如式(1)所示的函数表达式中乘积项间的公因子,由于除 NOT 运算符外, BED 所支持的其他运算符均为双目运算符,因此一种方法是识别出乘积项间大小为 2 的公因子 $\tilde{x}_i \tilde{x}_j$ ($\tilde{x}_i \in \{\tilde{x}_i, x_i\}$). 如果 $\tilde{x}_i \tilde{x}_j$ 被多个乘积项共享,那么在 BED 中,由 $\tilde{x}_i \tilde{x}_j$ 构建的运算符结点也会被多个父结点共享. 因此研究者提出一种基于共享度的变量配对方法对变量进行分组来实现变量排序。

对于任意一个变量 x_i ($1 \leq i < n$),统计其与每一个变量 x_j ($i < j \leq n$) 的组合 $\tilde{x}_i \tilde{x}_j$ 的共享度,根据共享度的大小对变量 x_i 和 x_j 进行配对。

共享度的计算方法为:遍历函数表达式中的乘积项,统计 $\tilde{x}_i \tilde{x}_j$ 的 4 种可能存在形式 $\tilde{x}_i \tilde{x}_j, x_i \tilde{x}_j, \tilde{x}_i x_j$ 和 $x_i x_j$ 的出现频率之和,并统计 $\tilde{x}_i \tilde{x}_j$ 的存在形式个数,以 $\tilde{x}_i \tilde{x}_j$ 为例,如果 $\tilde{x}_i \tilde{x}_j$ 在所有乘积项中至少出现一次,则 $\tilde{x}_i \tilde{x}_j$ 的存在形式个数加 1. 然后计算 $\tilde{x}_i \tilde{x}_j$ 关于其存在形式个数的平均出现频率,该平均出现频率即为变量对 (i, j) 的共享度. 如果 $\tilde{x}_i \tilde{x}_j$ 的 4 种可能存在形式均不在任何一个乘积项中出现,则其共享度为 0。

对于多输出函数,则先构建所有函数输出所包含的乘积项的并集. 算法 2 给出了基于变量对共享度的

变量分组算法.

算法 2 变量分组算法

```

Step1:  $h \leftarrow \emptyset, g \leftarrow \emptyset;$ 
Step2: for  $i$  from 1 to  $n-1$ 
Step3:   for  $j$  from  $i+1$  to  $n$ 
Step4:     遍历乘积项, 计算变量对  $(i, j)$  的共享度  $s_{i,j}$ ;
Step5:     if  $s_{i,j} \neq 0$  then  $h \leftarrow h \cup \{(i, j)\};$ 
Step6:   end for
Step7: end for
Step8: 根据共享度将  $h$  中的变量对按照降序排序;
Step9: for each  $(i, k) \in h$ 
Step10:  if  $(i$  和  $k$  均没有在  $g$  中出现) then
Step11:     $g \leftarrow g \cup \{(i, k)\};$ 
Step12:     $h \leftarrow h \setminus \{(i, k)\};$ 
Step13:  end if
Step14: end for
Step15: for each  $(i, k) \in h$ 
Step16:  if  $(i$  没有在  $g$  中出现) then  $g \leftarrow g \cup \{(i, -)\};$ 
Step17:  if  $(k$  没有在  $g$  中出现) then  $g \leftarrow g \cup \{(k, -)\};$ 
Step18: end for

```

算法 2 的 Step9 ~ Step14 依据变量对间没有相同元素的原则按共享度的大小将变量对 (i, k) 依次添加到集合 g 中; Step15 ~ Step18 则依据类似的原则将剩余的变量添加到集合 g 中, 其中的 $(i, -)$ 表示变量 x_i 不与其他变量进行配对.

算法 2 可以得到变量的一个分组 g , 其中变量对 (i, k) 中变量的出现次序则构成了一个变量顺序. 使用算法 2, 设计如算法 3 所示改进的基于 BED 的可逆电路综合算法.

对于一个多输出函数 $F = \{f_1, f_2, \dots, f_o\} (o \geq 1)$, o 为其原始输出数, 假设使用 GL 表示综合 F 所得可逆电路的垃圾线数, 则有 $GL = QB - o^{[19]}$. 可见, 如果一种可逆电路综合方法能够减少综合 F 所得电路的量子位数, 则也能减少所得电路的垃圾线数, 反之亦然.

算法 3 改进的可逆电路综合算法

```

Step1:  $QC \leftarrow 0, QB \leftarrow 0;$ 
Step2: 读取函数的 PLA 格式网表;
Step3: 运行算法 2 得到变量分组  $g$ ;
Step4: 按照  $g$  中的变量对  $(i, k)$  的顺序依次解析函数的每一个乘积项, 构建 BED;
Step5: 遍历 BED, 对于每一个结点  $v_j$ , 若  $v_j$  是变量结点则将其综合为一条线,  $QB \leftarrow QB + 1$ ; 若  $v_j$  是运算符结点, 则按节 3.1 所述方法对其进行综合, 级联可逆电路,  $QC \leftarrow QC + QC(v_j), QB \leftarrow QB + AQB(v_j)$ .

```

4 实验结果及分析

算法 1 和算法 3 均采用 C 语言实现, 在 Linux 下使

用 gcc 编译器编译, 并在配置为 Intel Core i7-6500U CPU 8GB RAM 的个人计算机上对算法进行验证.

由节 3.1 可知, 在 BED 中使用 XOR 和 BIIMP 结点有利于减少量子位数, 因此本文先使用 EXORCISM-4 工具^[24]利用“q0”参数对函数进行 ESOP 最小化, 得到函数的近优 ESOP 表示, 然后再使用算法 1 和算法 3 进行可逆电路综合, 算法 1 和算法 3 的 Step2 读取的是函数的 ESOP 表示的 PLA 格式网表.

4.1 变量分组算法效果验证

算法 1 使用默认变量顺序构建 BED, 算法 3 使用由本文的变量分组算法 (算法 2) 得到的变量分组构建 BED, 将算法 3 所得结果与算法 1 所得结果进行比较, 即可得出本文的变量分组算法对基于 BED 表示模型进行可逆电路综合所得结果的影响. 因此, 分别使用算法 1 和算法 3 对 28 个 PLA 格式的 MCNC 和 RevLib 函数进行可逆电路综合, 结果如表 2 所示, 其中 t 表示算法 3 以秒为单位的运行时间, “改善”表示相对于算法 1 的结果, 算法 3 结果的量子成本、量子位数和垃圾线数减少的百分比.

由表 2 数据可以看出, 与默认变量顺序相比, 对于超过半数的函数, 由算法 2 所得变量分组构成的变量顺序能有效降低由 BED 综合所得可逆电路的量子成本、量子位数以及垃圾线数, 这 3 个指标被改善的比例分别高达 47.85% (函数 apex5)、58.71% 和 59.98% (函数 seq). 从平均角度看, 本文的变量分组算法将由 BED 综合所得可逆电路的量子成本、量子位数和垃圾线数分别降低了 34.76%、40.35% 和 44.68%. 这验证了本文变量分组算法的有效性.

此外, 由表 2 可以看出, 对于大部分函数算法 3 都可在小于 0.01 秒的时间内完成可逆电路的综合; 函数 apex5、i6 和 i7 的输入变量数 n 均大于 100, 算法 3 的运行时间也较短, 最长时间仅为 0.42 秒 (函数 i7). 可见算法 3 具有较高的时间效率和很好的可扩展性.

4.2 与基于 DD 表示模型的可逆电路综合方法比较

与文献[2, 15 ~ 19, 25, 26]使用 DD 作为函数表示模型综合可逆电路的方法类似, 本文基于 BED 的可逆电路综合方法也属于结构综合方法, 因此选择与这些文献中的方法进行比较. 由于节 4.1 已经将算法 3 的结果与算法 1 的结果进行了比较, 因此下面仅将算法 3 的结果与这些文献中的结果进行比较, 结果如表 3 所示. 表 3 中的“现有基于 DD 表示模型的综合方法的结果”来自于文献[2, 15 ~ 19, 25, 26], 该结果是这些文献中从量子位数角度来看的最优结果, 即量子位数最优结果. 其中“来源”表示该结果来自于哪一篇文献, “决策图”表示该文献采用何种决策图模型综合可逆电路,

表 2 变量分组算法验证结果

函数	算法 1			算法 3				改善(%)		
	QC	QB	GL	QC	QB	GL	<i>t</i> /s	QC	QB	GL
5mod5	72	20	14	60	17	11	<0.01	16.67	15.00	21.43
aj-e11	69	16	12	67	16	12	<0.01	2.90	0	0
apex5	6152	1470	1382	3208	732	644	0.32	47.85	50.20	53.4
bw	463	82	54	391	65	37	0.01	15.55	20.73	31.48
cps	4446	752	643	3068	410	301	0.22	30.99	45.48	53.19
dc1	100	21	14	99	19	12	<0.01	1	9.52	14.29
decod24	19	6	2	19	6	2	<0.01	0	0	0
ex1	4	5	4	4	5	4	<0.01	0	0	0
ex-1	18	6	3	18	6	3	<0.01	0	0	0
ex5p	1725	322	259	1344	215	152	0.06	22.09	33.23	41.31
graycode6	5	6	0	5	6	0	<0.01	0	0	0
ham15	193	45	30	198	45	30	0.01	-2.59	0	0
ham3	17	6	3	13	6	3	<0.01	23.53	0	0
i6	738	279	212	738	279	212	0.22	0	0	0
i7	1055	405	338	1051	404	337	0.42	0.38	0.25	0.30
inc	317	65	56	313	63	54	<0.01	1.26	3.08	3.57
majority	54	17	16	29	11	10	<0.01	46.30	35.29	37.50
mod10	53	15	11	51	14	10	<0.01	3.77	6.67	9.09
mod5adder	211	47	41	166	37	31	<0.01	21.33	21.28	24.39
mod5d2	33	10	5	27	10	5	<0.01	18.18	0	0
one-two-three	28	8	5	28	8	5	<0.01	0	0	0
peres	8	4	1	8	4	1	<0.01	0	0	0
rd32	14	6	4	14	6	4	<0.01	0	0	0
sao2	630	149	145	459	103	99	0.01	27.14	30.87	31.72
seq	8567	1664	1629	4640	687	652	0.06	45.84	58.71	59.98
sqr6	295	63	51	280	58	46	<0.01	5.08	7.94	9.80
urf5	1675	345	336	1291	246	237	<0.01	22.93	28.70	29.46
xor5	4	5	4	4	5	4	<0.01	0	0	0
平均	963.04	208.54	188.36	628.32	124.39	104.21		34.76	40.35	44.68

“改善”表示相对于现有基于 DD 表示模型的综合方法的量子位数最优结果,算法 3 结果的量子成本、量子位数和垃圾线数减少的百分比。

由表 2 和表 3 可以看出,与现有基于 DD 表示模型的综合方法的量子位数最优结果相比,对于除 ex-1 外的所有函数,算法 3 能够获得更低的量子成本,并且量子成本被改善的比例高达 67.10% (函数 i6)。其中有 1 个函数(函数 xor5)在获得相同量子位数和垃圾线数的前提下降低了量子成本;有 10 个函数,可以在获得更少

的量子位数以及垃圾线数的前提下降低量子成本。从平均角度来看,算法 3 将可逆电路的量子成本降低了 45.67%,将量子位数和垃圾线数分别减少了 13.47% 和 15.67%。

对表 3 中函数的 ESOP 表示进行分析发现,对于那些乘积项中文字较少且乘积项间的结构相似性较高的函数,与其他基于 DD 表示模型的综合方法相比,算法 3 能够降低综合所得可逆电路的量子成本、量子位数以及垃圾线数。

表 3 与现有基于 DD 表示模型的综合方法的量子位数最优结果比较

函数	现有基于 DD 表示模型的综合方法的结果					改善(%)		
	QC	QB	GL	来源	决策图	QC	QB	GL
5mod5	143	14	8	文献[16]	PDD	58.04	-21.43	-37.50
bw	618	71	43			36.73	8.45	13.95
decod24	20	5	1	文献[15]	BDD	5	-20	-100
ham3	18	4	1			27.78	-50	-200
rd32	20	5	3			30	-20	-33.33
one-two-three	33	7	4			15.15	-14.29	-25
xor5	8	5	4			50	0	0
apex5	9023	1029	941	文献[18]	KFDD	64.45	28.86	31.56
ex5p	1525	202	139			11.87	-6.44	-9.35
seq	6214	828	793			25.33	17.03	17.78
urf5	1679	207	198			23.11	-18.84	-19.70
cps	5677	619	510	文献[19]	KFDD	45.96	33.76	40.98
dc1	141	21	14			29.79	9.52	14.29
ex1	6	6	5			33.33	16.67	20
inc	442	56	47			29.19	-12.50	-14.89
majority	37	10	9			21.62	-10	-11.11
mod10	60	12	8			15	-16.67	-25
sao2	562	74	70			18.33	-39.19	-41.43
sqr6	304	48	36			7.89	-20.83	-27.78
aj-e11	84	15	11	文献[26]	BDD	20.24	-6.67	-9.09
ex-1	17	5	2			-5.88	-20	-50
graycode6	15	11	5			66.67	45.45	100
mod5d2	30	11	6			10	9.09	16.67
peres	11	5	2			27.27	20	50
ham15	228	40	25	文献[17]	KFDD	13.16	-12.50	-20
i6	2243	280	213	文献[25]	BDD	67.10	0.36	0.47
i7	2953	403	336			64.41	-0.25	-0.30
mod5adder	269	32	26	文献[2]	BBDD	38.29	-15.63	-19.23
平均	1156.43	143.75	123.57			45.67	13.47	15.67

对于表 3 中的这些函数,在现有基于 DD 表示模型的综合方法所得结果中,有 5 个函数的量子成本最优结果与量子位数最优结果不同,表 4 给出了这 5 个函数的量子成本最优结果,以及算法 3 的结果相对于这些结果所改善的百分比。

由表 2 和 4 可以看出,与现有基于 DD 表示模型的综合方法的量子成本最优结果相比,对于这 5 个函数,算法 3 也能够降低综合所得可逆电路的量子成本,并且对于除函数 ex5p 和 rd32 外的其他 3 个函数,算法 3

也能够减少综合所得可逆电路的量子位数和垃圾线数。对于函数 ham15 和 rd32,尽管与表 3 中现有方法的量子位数最优结果相比,算法 3 结果的量子位数和垃圾线数有所增加,但是与现有方法的量子成本最优结果相比,对于函数 ham15,算法 3 在将量子成本降低 12.39% 的前提下,将量子位数和垃圾线数分别减少了 69.59% 和 77.44%,对于函数 rd32,算法 3 在不增加量子位数和垃圾线数的前提下,将量子成本降低了 12.50%。

表 4 与现有基于 DD 表示模型的综合方法的量子成本最优结果比较

函数	现有基于 DD 表示模型的综合方法的结果					改善(%)		
	QC	QB	GL	来源	决策图	QC	QB	GL
ex5p	1388	206	143	文献[26]	BDD	3.17	-4.37	-6.29
bw	581	78	50	文献[19]	KFDD	32.70	16.67	26
rd32	16	6	4			12.50	0	0
xor5	6	6	5			33.33	16.67	20
ham15	226	148	133	文献[16]	PDD	12.39	69.59	77.44

由以上分析可以看出,与现有基于 DD 表示模型的综合方法相比,对于几乎所有的实验所用函数,算法 3 能够改善综合所得可逆电路的量子成本,并且在许多情况下还能减少量子位数和垃圾线数。

另外,与现有基于 DD 表示模型的综合方法类似,本文所提出的综合方法也将不可逆函数的嵌入结合到了可逆电路的综合过程之中,并且具有较高的时间效率和很好的可扩展性。

5 结语

采用复杂度较低的决策图表示形式有利于降低由决策图综合所得可逆电路的成本。本文提出了基于 BED 表示模型的可逆电路综合方法,该方法采用深度优先后序方式遍历 BED 结点,通过将结点函数综合为可逆子电路实现可逆电路的综合。为进一步降低 BED 的复杂度,从而降低由 BED 综合所得可逆电路的量子成本和量子位数,本文还提出了一种基于共享度对函数输入变量进行分组实现变量排序的方法,并根据所得变量分组依次解析函数表达式中的乘积项来构建 BED。一组 MCNC 和 RevLib 函数的可逆电路综合结果验证了所提出方法的有效性。与现有基于 DD 表示模型的可逆电路综合方法所得结果相比,本文的可逆电路综合方法使量子成本最高降低了 67.10%、量子位数最大减少了 69.59%。

由 BED 综合所得可逆电路的成本与变量顺序有关,如何获得更好的变量顺序是本文研究者下一步的一项研究工作。此外,除本文中的混合极性 Peres 门外,还有一些可逆逻辑门(如 Double Peres 门等)也具有较低的量子成本,如何利用这些可逆门进一步降低由 BED 综合所得可逆电路的成本将是本文研究者下一步的研究重点。

参考文献

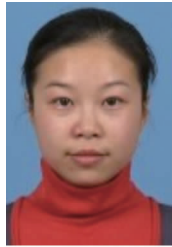
- [1] Wille R, Drechsler R. Towards a Design Flow for Reversible Logic[M]. New York: Springer, 2010.
- [2] Chattopadhyay A, Littarru A, Amaru L, et al. Reversible logic synthesis via biconditional binary decision diagrams [A]. Proceedings of the 45th International Symposium on Multiple-Valued Logic[C]. Waterloo: IEEE Press, 2015. 2-7.
- [3] Bennett C H. Logical reversibility of computation[J]. IBM Journal of Research and Development, 1973, 17: 525-532.
- [4] Shukla V, Singh O P, Mishra G R, et al. Reversible realization of n-bit arithmetic circuit for low power loss ALU applications[J]. Procedia Computer Science, 2018, 125: 847-854.
- [5] Nielsen M A, Chuang I L. Quantum Computation and Quantum Information[M]. Cambridge: Cambridge University Press, 2010.
- [6] Abdessaied N, Drechsler R. Reversible and Quantum Circuits: Optimization and Complexity Analysis[M]. Bremen: Springer Press, 2016.
- [7] Zulehner A, Wille R. One-pass design of reversible circuits: combining embedding and synthesis for reversible logic[J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2018, 37(5): 996-1008.
- [8] Yang G, Xie F, Hung W N N, et al. Realization and synthesis of reversible functions[J]. Theoretical Computer Science, 2011, 412(17): 1606-1613.
- [9] 陈汉武, 李文骞, 阮越, 等. 基于汉明距离递减变换的可逆逻辑综合算法[J]. 计算机学报, 2014, 37(8): 1839-1845.
Chen Hanwu, Li Wenqian, Ruan Yue, et al. A synthesis algorithm of reversible logic circuit based on the decreasing transform of hamming distance[J]. Chinese Journal of Computers, 2014, 37(8): 1839-1845. (in Chinese)
- [10] Lin C-C, Jha N K. RMDDS: Reed-Muller decision diagram synthesis of reversible logic circuits[J]. ACM Journal on Emerging Technologies in Computing Systems, 2014, 10(2): 14:1-14:25.
- [11] Soeken M, Tague L, Dueck G W, et al. Ancilla-free synthesis of large reversible functions using binary decision diagrams[J]. Journal of Symbolic Computation, 2016, 73: 1-26.
- [12] Soeken M, Wille R, Keszocze O, et al. Embedding of large Boolean functions for reversible logic[J]. ACM

- Journal on Emerging Technologies in Computing Systems, 2015, 12(4): 41:1 – 41:26.
- [13] 卜登立. 基于 ESOP 最大加权输出相容类的可逆电路综合方法[J]. 电子学报, 2018, 46(8): 1866 – 1875.
Bu Dengli. Reversible circuit synthesis method based on maximum weighted output-compatibility class of ESOP [J]. Acta Electronica Sinica, 2018, 46(8): 1866 – 1875. (in Chinese)
- [14] Drechsler R, Wille R. Synthesis of reversible circuits using decision diagrams [A]. Proceedings of the International Symposium on Electronic System Design [C]. Kolkata, India; IEEE Press, 2012. 1 – 5.
- [15] Krishna M, Chattopadhyay A. Efficient reversible logic synthesis via isomorphic subgraph matching [A]. Proceedings of the IEEE 44th International Symposium on Multiple-Valued Logic [C]. Bremen, Germany: IEEE Press, 2014. 103 – 108.
- [16] Pang Y, Yan Y, Lin J, et al. An efficient method to synthesize reversible logic by using positive Davio decision diagrams [J]. Circuits, Systems, and Signal Processing, 2014, 33(10): 3107 – 3121.
- [17] 王友仁, 沈先坤, 周影辉. 基于 KFDD 的可逆逻辑电路综合设计方法 [J]. 电子学报, 2014, 42(5): 1025 – 1029.
Wang Youren, Shen Xiankun, Zhou Yinghui. Synthesis design method of reversible logic circuit based on Kronecker functional decision diagram [J]. Acta Electronica Sinica, 2014, 42(5): 1025 – 1029. (in Chinese)
- [18] Schonborn E, Datta K, Wille R, et al. Optimizing DD-based synthesis of reversible circuits using negative control lines [A]. Proceedings of the 17th International Symposium on Design and Diagnostics of Electronic Circuits & Systems [C]. Warsaw; IEEE Press, 2014. 129 – 134.
- [19] Law J J, Rice J E. Line reduction in reversible circuits using KFDDs [A]. Proceedings of the 2015 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing [C]. Victoria, BC, Canada; IEEE Press, 2015. 113 – 118.
- [20] Andersen H R, Hulgaard H. Boolean expression diagrams [J]. Information and Computation, 2002, 179(2): 194 – 212.
- [21] Hagjam F Z, Moraga C. RIMEP2: evolutionary design of reversible digital circuits [J]. ACM Journal on Emerging Technologies in Computing Systems, 2014, 11(3): 27:1 – 27:23.
- [22] Wille R, Große D, Teuber L, et al. RevLib: an online resource for reversible functions and reversible circuits [A]. Proceedings of the 38th International Symposium on Multiple-Valued Logic [C]. Dallas, TX; IEEE Press, 2008. 220 – 225.
- [23] Hansen J P, Sekine M. Synthesis by spectral translation using Boolean decision diagrams [A]. Proceedings of the 33rd Design Automation Conference [C]. Las Vegas, NV, USA; IEEE Press, 1996. 248 – 253.
- [24] Mishchenko A, Perkowski M. Fast heuristic minimization of exclusive-sums-of-products [A]. Proceedings of Reed-Muller Workshop [C]. Mississippi, USA; IEEE Press, 2001. 242 – 250.
- [25] Wille R, Drechsler R. BDD-based synthesis of reversible logic for large functions [A]. Proceedings of the 46th ACM/IEEE Design Automation Conference [C]. San Francisco, CA, USA; IEEE Press, 2009. 270 – 275.
- [26] Wille R, Drechsler R. Effect of BDD optimization on synthesis of reversible and quantum logic [J]. Electronic Notes in Theoretical Computer Science, 2010, 253(6): 57 – 70.

作者简介



卜登立 男, 1975 年出生, 河北定州人. 博士, 副教授, 中国电子学会高级会员, 主要研究领域为电路设计与优化、可逆逻辑综合、量子电路综合、启发式优化算法.
E-mail: bodengli@163.com



郭 鸣 女, 1978 年出生, 江西吉安人. 博士, 副教授, 主要研究领域为电路设计与优化.
E-mail: guoming@jgsu.edu.cn