

基于布尔可满足性的精确逻辑综合综述

储著飞* 潘鸿洋

(宁波大学信息科学与工程学院 宁波 315211)

摘要: 逻辑综合是电子设计自动化(EDA)的重要步骤,随着算力逐渐提升和新的计算范式不断涌现,传统基于全局启发式算法的逻辑综合面临新的挑战。启发式算法面临的主要问题是得到一个次优解,随着算力的提升,逻辑优化越来越追求精确解而不满足于次优解。该文首先简述逻辑函数表达方法和布尔可满足性(SAT)问题;其次针对精确综合的算法、编码等方面介绍了在布尔逻辑网络的面积优化和深度优化方面的精确综合研究进展;最后对精确综合的未来发展趋势进行讨论。

关键词: 逻辑综合; 精确综合; 布尔可满足性; 多数逻辑门

中图分类号: TN47

文献标识码: A

文章编号: 1009-5896(2023)01-0014-10

DOI: 10.11999/JEIT220391

Survey on Exact Logic Synthesis Based on Boolean SATisfiability

CHU Zhufei PAN Hongyang

(Faculty of Electrical Engineering and Computer Science (EECS), Ningbo University, Ningbo 315211, China)

Abstract: Logic synthesis is a critical step in the Electronic Design Automation(EDA). Traditional global heuristic-based logic synthesis has many challenges as computing power keeps increasing and new computing paradigms emerge. There is a problem with heuristic algorithm in a suboptimal solution. As computing power improving, logic optimization is increasingly pursuing exact solutions rather than suboptimal solutions. First, the logic representations and the Boolean SATisfiability(SAT) problem are briefly described. Then, the research progress of exact synthesis in area optimization and depth optimization of Boolean logic network at two aspects, exact synthesis algorithm and encoding, are introduced. Finally, the future trends in exact synthesis are discussed.

Key words: Logic synthesis; Exact synthesis; Boolean SATisfiability (SAT); Majority logic

1 引言

电子设计自动化(Electronic Design Automation, EDA)工具及其设计方法学是集成电路(芯片)设计的核心,EDA工具通过提供自动化设计过程帮助芯片设计师设计开发,主要分为前端设计和后端设计,其中,逻辑综合与优化是现代EDA流程中的关键组成部分,属于前端设计的核心。逻辑综合是将用户设计的数字电路转换为工艺库网表的过程,输入通常是寄存器传输级硬件描述语言(Register Transfer Level, RTL)、工艺库文件和设计约束文件,经过翻译、逻辑优化和工艺映射等步

骤输出工艺库网表。综合出的网表质量直接影响到后续的物理设计过程。

逻辑优化以逻辑表达为基础,在逻辑功能一致的前提下优化逻辑表达的逻辑深度、开关活动性、文字数(literals)或节点个数等,分别对应具体工艺下的性能(Performance)、功耗(Power)和面积(Area),即PPA优化。传统逻辑优化算法从其数学基础分类主要分为代数法和布尔法,其中代数法又称为多项式代数法,基于普通代数发展而来;布尔法则充分利用布尔代数区别于普通代数的特性,增强了对布尔空间的搜索能力。从算法寻优能力分类又可分为启发式算法和最优算法,前者用可接受的成本计算出一个次优解,而后者则不惜成本找到问题的最优解。超大规模数字集成电路在逻辑优化中因受制于规模约束,常采用全局启发式,局部最优算法的模式。在逻辑综合中,基于最优算法寻找最优逻辑表达式的过程称为精确逻辑综合(exact synthesis),最优综合和精确综合常互换使用^[1]。最优逻辑表达式指的是布尔逻辑函数的逻辑表达用到的

收稿日期: 2022-04-02; 改回日期: 2022-08-29; 网络出版: 2022-09-01

*通信作者: 储著飞 chuzhufei@nbu.edu.cn

基金项目: 国家自然科学基金(61871242), 专用集成电路与系统国家重点实验室开放研究课题基金(2021KF008)

Foundation Items: The National Natural Science Foundation of China (61871242), The Open research fund of State Key Laboratory of Application Specific Integrated Circuits and Systems (2021KF008)

节点数最少或者逻辑深度最小等。随着算力的提升, 逻辑优化越来越追求最优解而不满足于次优解。

精确综合采用的最优算法常通过约束求解器完成决策, 常用的求解器包括定理证明器、整数线性规划求解器、布尔可满足性(Boolean SATisfiability, SAT)求解器、可满足性模理论(Satisfiability Modulo Theory, SMT)求解器等。精确综合得到的结果可用于大规模数字电路的局部优化以及新兴计算范式的设计, 提升了综合优化的结果质量(quality of result)。因SAT求解器及其算法的发展, 近年来基于SAT编码及求解的精确综合受到广泛关注, 对专用集成电路(Application Specific Integrated Circuit, ASIC)和现场可编程门阵列(Field Programmable Gate Array, FPGA), 以及新兴计算范式如存算一体系统、近似计算、随机计算等带来新的优化机会。

本文对精确综合的SAT编码问题以及对应的精确综合应用进行综述, 首先对逻辑函数的表达方式和布尔可满足性问题等背景知识进行介绍; 第3节针对精确综合算法进行具体阐述; 第4节归纳总结了目前国内外常见的基于布尔可满足性的精确综合使用场景; 最后探讨了研究挑战和未来发展趋势。

2 背景

本节首先介绍逻辑函数的表达方式, 然后介绍SAT问题及常用的算法。

2.1 逻辑表示

如何表示逻辑函数是逻辑综合首先面临的问题, 逻辑函数从早期的真值表/卡诺图, 到“积之和(Sum Of Products, SOPs)”、2元决策图(Binary Decision Diagram, BDD)、发展到如今的有向无环图(Directed Acyclic Graph, DAG), 其可表达和处理的逻辑函数规模逐渐增大。

布尔链(Boolean chain)是DAG的一种, 每个顶点有 k 个输入, 执行 k 元逻辑运算。布尔链概念提出时 $k=2$ ^[2], 但 k 可推广到任意大的有限数, 且一个布尔链中的 k 是唯一的^[3]。其形式化定义如下, 给定一个 m 输出的函数 $f = \{f_1, f_2, \dots, f_m\}$, 每个输出函数的输入属于集合 $\{x_1, x_2, \dots, x_n\}$, 那么这些函数的布尔链可表示成一个 r 步(骤)序列 $(x_{n+1}, x_{n+2}, \dots, x_{n+r})$, 序列中每一步的输入都是前序已计算的步骤。以 $k=2$ 为例

$$x_i = x_{j(i)} \circ_i x_{k(i)} \quad (1)$$

其中, $n+1 \leq i \leq n+r$, $1 \leq j(i) < i$, $1 \leq k(i) < i$, $\circ_i$ 为 $2^{2^k}=16$ 种2元逻辑运算的一种。输出 $f_p = x_{l(p)}$

或者 $f_p = \bar{x}_{l(p)}$, 其中 $p \in [1, m]$, $l(p) \in [n+1, n+r]$, 注意至少有1个 $l(p) = n+r$ 。

以 $n=3, m=2, r=2$ 的布尔链为例, 3个输入为 $\{x_1, x_2, x_3\}$, 2个输出为 $\{f_1, f_2\}$, 布尔链序列为

$$x_4 = x_1 \wedge x_2 \quad (2)$$

$$x_5 = x_3 \oplus x_4 \quad (3)$$

且 $f_1 = x_4$, $f_2 = x_5$ 。因此 $l(1) = 4, l(2) = 5$ 。其DAG图形化示意图如图1所示, 其中‘ $\wedge$ ’为与门, $f = \{f_1, f_2\} = \{x_1x_2, x_1x_2 \oplus x_3\}$ 。

此外, 逻辑综合中常用的DAG还有AIG (AND-Inverter Graph), MIG (Majority-Inverter Graph)等。其中AIG与布尔链的区别是每个内部节点只能实现2输入‘与’操作, 而连接边包含原边和补边, 补边表示反相器运算; MIG则是对AIG的推广, 每个内部节点实现的是3输入“多数逻辑门”(majority-of-three)。因此, 通过限定布尔链的输入 k 以及内部节点的逻辑功能, 会衍生出不同的基于DAG的逻辑表示。

2.2 SAT问题

SAT问题是著名的判定问题, 给定描述问题的布尔公式, 判定是否存在1组或多组变量的取值组合, 能够使布尔公式取值为真, 如果存在使布尔公式取值为真的变量取值组合, 称该布尔公式是可满足的(SAT), 否则该布尔公式是不可满足的(UNSAT)。一个布尔变量或者其反变量称为文字, 若干文字的析取式又称为子句(clause), 所有子句的合取式则构成了描述问题的布尔公式, 即合取范式(Conjunction Normal Form, CNF)。

以 $x_2(x_1 \vee \bar{x}_2 \vee x_3)(\bar{x}_1 \vee x_3)$ 为例, 该CNF包含3个子句, 分别是 $x_2, (x_1 \vee \bar{x}_2 \vee x_3), (\bar{x}_1 \vee x_3)$, 其中 x_2 是单文字子句, 要使该CNF为真, 则每个子句必须为真, 此例中 x_2 只有赋值为真才能满足约束。经过推理, x_3 为真, x_1 任意赋值即可使CNF为真。将问题转换为CNF描述的过程称为SAT编码, 求解CNF的过程称为SAT求解算法, 给定CNF输入获得SAT判定结果的工具又称为SAT求解器。可见

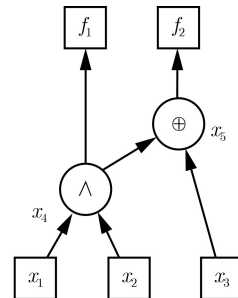


图1 布尔链示意图

SAT求解器的核心是对CNF的处理和求解算法,一般问题通过SAT编码调用现成的SAT求解器即可获得结果,因此很多SAT求解器的应用难点在于如何将问题进行高效的SAT编码。

SAT问题还有很多变种,比如 k -SAT问题限定子句中的文字数不超过 k ,最大SAT(max-SAT)则努力找到一个能满足最多子句的赋值,全解SAT(all-SAT)则需要找到所有可能的赋值。较常见的SAT求解器包括 Chaff, MiniSAT等。

3 精确综合

相比于基于启发式算法的逻辑综合,精确逻辑综合的研究则要少得多,其中最难突破的问题就是计算的复杂度高,需要依靠枚举完成推理和决策。奎因-麦克拉斯基(Quine-McCluskey)算法是知名的2级逻辑SOP最小化算法^[4],因受制于计算复杂度,后来发展的启发式优化方法Espresso更受欢迎。精确综合需要给定1个或者1组布尔逻辑函数 $\mathcal{F}$ 、用来表示逻辑函数的逻辑门集合/逻辑原语集合 $\mathcal{L}$ 以及成本函数 C 。

从精确综合所采用的方法来看,传统的精确综合包含以下3种方式^[5]。

(1) 任意逻辑函数可通过函数分解成逻辑表达式,一个典型的例子是香农分解(Shannon decomposition),函数 $f = xF_x + \bar{x}F_{\bar{x}}$,其中 F_x 和 $F_{\bar{x}}$ 为函数 f 关于变量 x 的正/负辅因数(cofactor), $F_x = f(x=1)$, $F_{\bar{x}} = f(x=0)$ 。通过对逻辑函数递归调用香农分解可得到逻辑表达式。此外,还有非相交分解(Disjoint-Support Decomposition, DSD)、对分分解(Bi-decomposition)等分解手段。通过不同的分解方式得到逻辑表达式从中寻优得到最优解。

(2) 给定逻辑操作符号集合 L ,由 n 个逻辑门组成的逻辑网络个数是有限的,通过枚举逻辑网络,并对逻辑网络节点赋操作符,可得到不同的逻辑函数,得到的逻辑函数与综合的逻辑函数进行比对,如果一致则找到最优解,否则增大 n 进入下一次枚举。 n 从0开始,显然,随着 n 增大所枚举的逻辑网络个数非线性增长。

(3) 结合第(1)和第(2)种方法同时进行函数分解和逻辑网络枚举的混合方式寻优。

然而,这些方法所能处理的逻辑函数规模非常有限,只能通过枚举来证明最优性。近年来,随着算力的提高以及理论计算机科学的发展,利用约束求解器,特别是SAT求解器进行决策和判定的精确综合方法受到广泛关注。基于SAT的精确综合在2007年计算机辅助设计中的形式化方法(Formal

Methods in Computer-Aided Design, FMCAD)会议上由MiniSAT作者Eén^[6]提出。随后,Knuth^[2]使用了不同的CNF编码进行精确综合,但仅限于2输入算子最优布尔链计算,这些算法都旨在找到面积最优(布尔链节点数最少)的布尔链。Soeken等人^[7]将它们扩展到综合逻辑深度最优的布尔链,使用SAT求解器来检查是否存在约束下实现给定功能的最少逻辑层级的布尔逻辑网络。故基于SAT的精确综合的任务是在大小或深度方面为给定的输入约束找到最优逻辑表达。

3.1 精确综合算法流程

图2(a)所示为求解最优面积的精确综合算法流程图,核心思想是通过约束编码并调用求解器回答问题“是否存在 r 个属于集合 $\mathcal{L}$ 的逻辑门构成的逻辑网络实现待综合的布尔函数集合 $\mathcal{F}$?”,成本函数 $C=r$ 。因此 r 的初值为0,这主要是考虑到一些待综合的函数是常数0或1、原始输入(primary input)或者其反变量等,此时不需要逻辑门,例如 $\mathcal{F}=\{1, x_1\}$, x_1 为原始输入,则待综合的函数就是不需要逻辑门的简单情况。在给定的 r 变量约束下通过产生基于CNF的SAT编码 $\mathcal{F}_r$,并将其交给SAT求解器计算,若SAT求解器返回可满足解,则找到了实现待综合函数的最优逻辑网络;否则将 r 增1,并再次编码求解指导SAT返回可满足解。因此能确保得到结果是最优的,即用最少逻辑门实现待综合逻辑函数。给定 r 的上界或者限定SAT求解器的时间都能使算法提前终止。

从算法的流程看出,在 r 变量和逻辑门集合 $\mathcal{L}$ 的约束下进行SAT编码是最重要的步骤。此外,算法的复杂度与 r 的大小密切相关,当 r 较小时,SAT编码的变量个数和子句个数相对较少,因此SAT求解器往往能快速返回结果;如果一个函数很复杂,需要较多的 r 个逻辑门才能实现,显然SAT求解器的求解难度会随着SAT编码变量的个数以及子句个数增加而增大,例如假定最优解 $r=10$,那么在 $r=0\sim 9$ 的自底向上递增过程中SAT都在努力证明这些 $\mathcal{F}_r$ 是不满足的(UNSAT)。

相比于自底向上增加 r 引起的无效搜索,文献^[8]提出了自顶向下降低 r 的综合流程如图2(b)所示。首先通过函数分解获得待综合函数节点个数上界 r ,尝试调用SAT求解器回答问题“是否存在 $r=r-1$ 个属于集合 $\mathcal{L}$ 的逻辑门构成的逻辑网络实现待综合的布尔函数集合 $\mathcal{F}$?”,如果是不满足的,则说明 $r+1$ 已经是最优值;否则继续减小 r 直到SAT求解器返回UNSAT。同理,如何获得一个较为紧凑的 r 上界至关重要。

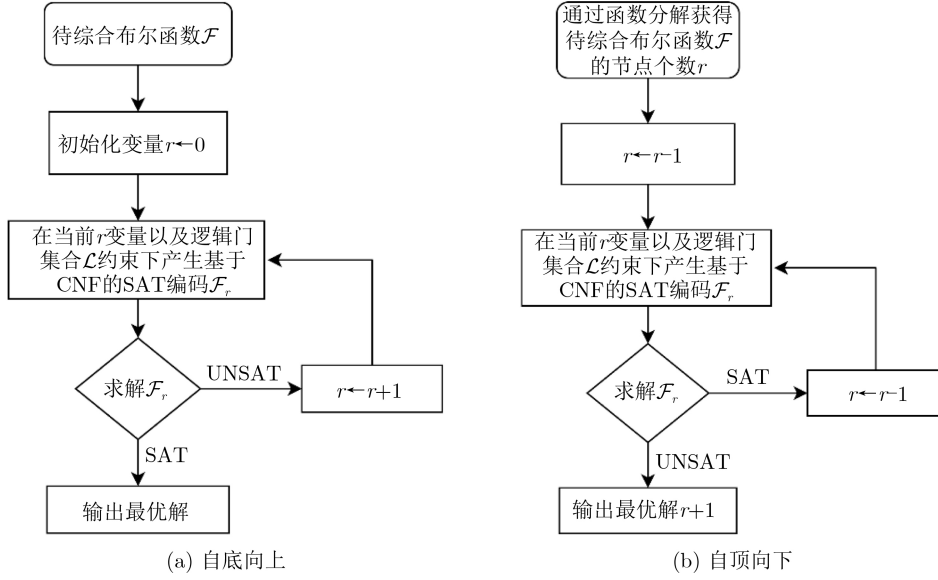


图2 基于SAT的精确综合流程图

3.2 SAT编码

SAT编码的两个核心工作在于定义布尔变量，以及基于这些定义的布尔变量添加约束。定义逻辑函数 $f(x_1, x_2, \dots, x_n) = f(0, 0, \dots, 0) = 0$ 为常规函数 (normal function)，一个布尔链中的所有步骤若都是常规函数，则称该布尔链为常规布尔链。例如一个布尔链若由“与/或”组成，则是常规布尔链；相比若布尔链包含“与非/或非”，则不是常规布尔链。一个非常规布尔函数可以通过添加反相器的方式使其成为常规函数的反相，例如 $f(a, b) = \overline{a + b}$ ，虽然 f 是非常规函数，但是 f 是常规函数。本节以 Knuth^[2]提出的针对2输入常规布尔链CNF编码为例加以说明。

(1) 变量定义。若待综合的逻辑函数原始输入数为 n ，原始输出数为 m ，当前待编码的逻辑门个数是 r ，对于 $1 \leq h \leq m, n < i \leq n + r$ ，且 $0 < t < 2^n$ ，分别定义如下布尔变量：

- (a) x_{it} ，表示逻辑门 x_i 真值表的第 t 位。
- (b) g_{hi} ，表示逻辑门 x_i 是逻辑函数的第 h 个输出。
- (c) s_{ijk} ，表示逻辑门 x_i 的输入是 x_j 和 x_k ，其中 $1 \leq j < k < i$ 。
- (d) f_{ipq} ，表示逻辑门 x_i 的输入布尔赋值为 (p, q) 时的输出布尔值。

为了降低变量个数，其中 f_{ipq} 中的 f_{i00} 不会出现，因为常规布尔链中 $f_i(0, 0) = 0$ 。

(2) 约束。约束包含基本约束和扩展约束，其中基本约束确保能够得到正确结果，而扩展约束通过逻辑网络的特性加入了对称性破坏 (symmetry breaking) 规则以减少搜索空间^[2]，受篇幅限制，本文仅介绍基本约束。

基本约束中，最核心的是确保布尔链中的逻辑门完成正确的逻辑运算，该约束又称为主约束子句 (main clauses)。对于 $0 \leq a, b, c \leq 1$ ，且 $1 \leq j < k < i$ ，主子句约束定义为

$$((s_{ijk} \wedge (x_{it} \oplus \bar{a}) \wedge (x_{jt} \oplus b) \wedge (x_{kt} \oplus \bar{c})) \rightarrow (f_{ibc} \oplus \bar{a})) \quad (4)$$

式(4)为蕴涵逻辑算式，即 $f = y_1 \rightarrow y_2 = \bar{y}_1 + y_2$ ，可将其转化为CNF形式，即

$$((\bar{s}_{ijk} \vee (x_{it} \oplus a) \vee (x_{jt} \oplus b) \vee (x_{kt} \oplus c)) \vee (f_{ibc} \oplus \bar{a})) \quad (5)$$

该约束可翻译为，若逻辑门 x_i 的输入分别是 x_j 和 x_k ，且 x_i, x_j 和 x_k 在真值表第 t 位的值分别为 a, b 和 c ，那么逻辑 x_i 一定执行的是逻辑运算 $b \circ_i c = a$ 。此处 a, b 和 c ，可控制变量的极性。

此外，若逻辑门 x_i 是输出，那么 x_i 的真值表一定是等于其中待综合的布尔函数，因此添加约束

$$\bar{g}_{hi} \vee (x_{it} \oplus g_h(t_1, t_2, \dots, t_n)) \quad (6)$$

其中， $(t_1, t_2, \dots, t_n)_2$ 是 t 的二进制编码。对于每一个输出函数，增加约束 $\bigvee_{i=r+1}^{n+r} g_{ht}$ 确保有一个步骤是输出。最后添加约束 $\bigvee_{k=1}^{i-1} \bigvee_{j=1}^{k-1} s_{ijk}$ 确保布尔链中每个步骤都有两个合法的输入。

(3) 示例。以图1所示的布尔链为例， x_{it} 变量根据真值表对每个逻辑门的全局函数进行编码，由于 $n = 3, m = 2, r = 2$ ，故逻辑门的编号属于区间 $(n, n + r]$ ，即 x_4 和 x_5 ，一共有 $2^n - 1 = 7$ 个真值表位，注意第0位因常规函数无需编码

$$\left. \begin{array}{l} t=7 \ 6 \ 5 \ 4 \ 3 \ 2 \ 1 \\ x_{4t} = 1 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \\ x_{5t} = 0 \ 1 \ 1 \ 1 \ 1 \ 0 \ 0 \end{array} \right\} \quad (7)$$

一共有4个 g_{hi} 变量，其中有2个赋值为1，表示函数

输入对应的逻辑门, 可知 $g_{14} = 1, g_{15} = 0, g_{24} = 0, g_{25} = 1$ 。一共存在9个选择变量, 赋值分别为

$$\left. \begin{array}{l} k=2 \ 3 \ 4 \\ s_{41k} = 1 \ 0 \\ s_{42k} = 0 \\ s_{51k} = 0 \ 0 \ 0 \\ s_{52k} = 0 \ 0 \\ s_{53k} = 1 \end{array} \right\} \quad (8)$$

最后, f_{ipq} 变量对与门或异或门的真值表进行编码为

$$\left. \begin{array}{l} (p, q) = (1, 1)(0, 1)(1, 0) \\ f_{4pq} = 1 \ 0 \ 0 \\ f_{5pq} = 0 \ 1 \ 1 \end{array} \right\} \quad (9)$$

4 基于SAT的精确综合研究进展

精确综合中如何高效进行SAT编码以便更快求解是关键, 精确综合可以在线(online)或者离线(offline)的形式应用到大规模集成电路综合中。既可应用到逻辑综合中的逻辑网络重构、布尔函数等价分类等场景中, 也可面向后摩尔时代新兴计算范式中的新型逻辑原语、设计约束开展综合优化。本节重点介绍精确综合当前的研究进展。

4.1 SAT编码

现有SAT编码仅定义了基本约束, SAT求解器在进行状态空间搜索时, 需要同时完成两个任务: 搜索逻辑网络的结构和搜索每个节点的逻辑操作符, 因此随着逻辑函数所需的计算步骤(节点数)增多, SAT求解器返回解的时间相对较长。若逻辑网络结构已知则使SAT编码问题变得容易, 因此通过添加一种基于DAG拓扑族的约束, 对逻辑网络的结构和其节点的功能进行编码, 可以极大程度上限制SAT求解器的搜索空间, 实现更快找到最优替换网络。

Haaswijk等人^[9]使用基于DAG拓扑结构枚举的精确综合算法限制了SAT求解器的搜索空间。由于拓扑结构的数量随着逻辑门数量的增加而快速增长, 作者由此提出一种全新的拓扑结构—布尔围栏(Boolean fence), 如图3所示, 对于拥有4个节点的逻辑网络对应的DAG族进行分类, 每个围栏对应于一个总节点分布相同的DAG族。通过枚举和使用该DAG族来约束基于SAT的精确综合编码, 加速了基于精确综合的各种逻辑优化算法。文献^[3]在此基础上, 重新审视基于SAT的精确综合的编码问题, 收集拓扑族中的结构集, 提出了3种优化来加速编码并减少运行时间。分别为: 量化CNF编码方案之间的差异及其对运行时间的影响; 证明对称破缺约束的积极影响; 使用多线程SAT求解器对拓扑信息进行计算。结果表明, 使用DAG拓扑族可以有效地减少精确综合运行时间和提高精确综合的应

用范围, 同时并行化精确综合会大幅减少运行时间, 这些改进对使用精确综合的各种逻辑优化算法有直接影响。

现有的SAT编码较多采用的逻辑门集合 $\mathcal{L}$ 是2输入逻辑门, 布尔函数的空间十分大, 其枚举或者计算可能难以处理。因此, 通过引入更多输入的多数逻辑门数据结构作为网络的逻辑表示, 在精确综合中使用这些原语会减少运行时间, 使最佳布尔链的逻辑重写性能得到大幅提升。

Soeken等人^[7]使用MIG作为基础逻辑表示, 减少了SAT的求解时间。MIG为同构逻辑表示并且逻辑表达能力强, 表达形式如图4(a)所示。由于MIG规则的数据结构, 可以丢弃输出极性变量并引入对称破坏规则。使用该算法, 能够改进基于查找表(LookUp Table, LUT)的工艺映射后的面积和延迟。在此基础上, Haaswijk等人^[10]将MIG扩展到功能缩减的MIG (Functionally Reduced MIGs, FRMIG), 改进了MIG逻辑优化和基于LUT的映射。文章引入了基于 k -LUT映射和精确 k -LUT分解的通用MIG优化方法, 可用于深度和面积优化。文献^[11]引入一种更紧凑的逻辑表示异或多数逻辑图(XOR-Majority Graphs, XMG), 可以实现更小的逻辑网络, 从而在面积优化中扩大使用精确综合的范围。具体方法是改进了子网络选择策略, 避免了枚举, 允许算法扩展到更大的子网络, 从而消除了预计算和存储所有精确解的需要。Chu等人^[12]通过引入M55输入多数逻辑门 (majority-of-five)算子计算操作数最少的布尔函数, 降低了SAT编码的难度, 从而加快SAT求解速度。作者利用5输入多数算子的对称性, 以符号编码的方法来表示节点的功能, 以此减少变量的数量。此外, 作者引入5输入多数反相器图(M_5 -Inverter Graphs, M_5IG)进行操作, 这

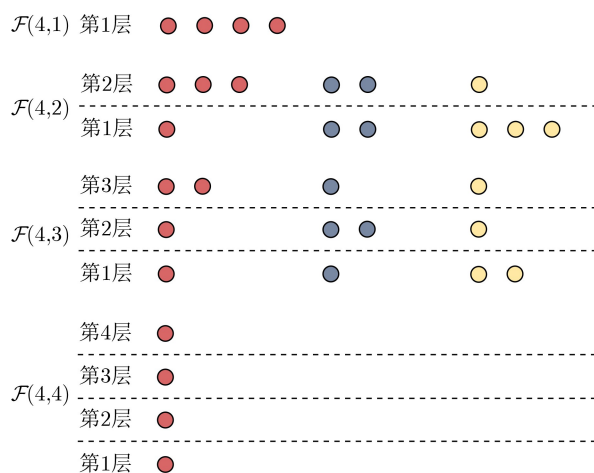


图3 围栏结构

是MIG的扩展,如图4(b)所示,其中“ $\langle \rangle$ ”为多数逻辑门,在图4(a)中为 M_3 ,在图4(b)中为 M_5 。

最后,Marakkalage等人^[13]使用基于SAT的精确综合分析在不同输入逻辑门中,用于同构逻辑表示的最优逻辑原语。通过对5输入和6输入布尔函数NPN等价类进行结果分析,证明3输入逻辑门在可表达性方面是最强大的。因此在基于SAT的精确综合中,我们更希望逻辑算子为3输入多数逻辑门。

4.2 逻辑函数等价分类及等价匹配

逻辑函数等价分类的目标是按照分类规则找到互相等价的一组布尔函数,使用一个具有代表性的函数来处理其他布尔函数。根据使用变换的不同,逻辑函数等价分类分为多种,研究最多的是否定置换否定(Negation-Permutation-Negation, NPN)等价分类,其中N-P-N分别代表对输入的补操作、对输入的置换操作和对输出的补操作。布尔函数等价匹配用来判定一个布尔函数与另一个布尔函数在功能上是否是等价的。NPN等价匹配是在给定的准则下确定两个逻辑函数是否属于同一个NPN类,用于将函数与已知的函数库进行匹配。在逻辑综合中,随着逻辑函数规模的不断扩大,对NPN等价匹配的可伸缩性和灵活性要求变高,在众多匹配方法中,基于SAT的匹配为综合问题编码为SAT公式提供了足够的自由度,具有较高的可伸缩性和灵活性。因此,将基于SAT的精确综合运用到NPN等价分类问题和快速NPN等价匹配算法中,在给定分类规则下找到最优布尔函数,构建更好的分类和更快的匹配等价布尔函数。

Haaswijk等人^[1,14]提出了一种以3输入运算符的组合复杂性为基础,对所有4和5输入函数进行分类的方法,同时使用基于SAT的精确综合,取消枚举所有布尔链的算法。该方法不仅可以用于计算NPN类和函数的数量,还可以用作精确综合算法的基础,基于将函数映射到其NPN类和NPN类到最优布尔链的两层预计算索引,如图5(a)所示。文献^[15]在此基础上,设计了类层次结构以更快速进行NPN分类,如图5(b)所示。在现有的算法中,所有的类都属于一个级别,并且函数直接与这些最终的NPN类匹配,而具有类的层次结构允许将函数匹配到第1级(L_1)的中间类,这些中间类被进一步匹配到来自第2级(L_2)的最终NPN类。该算法使用精确综合分别通过计算真值表和辅因子中1的数量来确定输出的极性和每个输入的极性。如果某些输入具有相同数量的1,则形成一个绑定组,并且它们的位置是不明确的。因此,在每个绑定组内使用SAT计算一系列否定和排列,从而最小化函数真值表中的值,以分类具有相同数量1的输入。

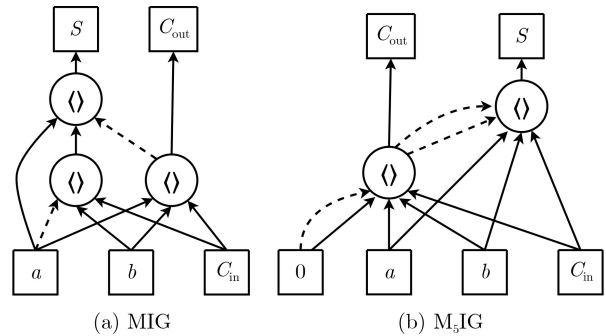


图4 全加器逻辑网络的不同多数逻辑表达

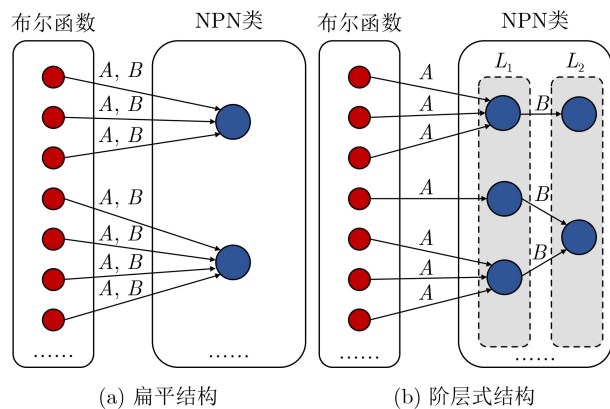


图5 NPN结构

Huang等人^[16]以真值表为基础,实现了完全指定布尔函数的基于NPN分类的快速布尔匹配算法。作者以真值表表示的6~16个输入布尔函数的精确NPN规范形式为基础,同时检测对称变量以增加算法的效率,并适度减少了内存使用量。由于使用了真值表,变量数量呈指数增长,该算法计算更多输入的函数会降低运行时间。对于更多输入的函数,Zhou等人^[17]使用了分层方法消除了重复计算,升级了变量对称性减少了要枚举的变换,准确地对多达16个输入的任何实用布尔函数进行分类。该快速NPN分类算法设计了移位辅因子签名(shifted-cofactor signatures),将相位枚举和置换枚举分开,并以成本感知规范形式最小化排列枚举成本。精确综合算法的输入大小限制是由真值表表示引起的,而该方法可以使用BDD来实现,对于实际应用出现的任何输入多达16个的布尔函数,该算法使精确分类问题首次可以得到有效解决。

4.3 布尔函数逻辑重构

在逻辑优化中,逻辑重写/重构(rewriting)通过对未优化逻辑网络进行模板模式匹配,用等效的最优逻辑网络进行替换,以达到优化节点数或逻辑深度的目的。逻辑重写一般通过切割枚举(cut enumeration)算法将给定逻辑网络划分为更小的子网络,对于任何子网络,在逻辑功能一致的前提下,

在其替代逻辑网络中选择一个最优的逻辑网络进行替换。由于SAT求解器在子问题中进行的决策更少,故求解子问题比求解原问题要快得多。因此,基于SAT的精确综合常用于逻辑重写的相关应用中,通过降低SAT求解器的计算复杂度,实现更快找到最优替换网络。

Soeken等人^[18]设计了一种名为“忙人综合”(Busy Man's Synthesis, BMS)的高效深度降低逻辑重写算法,根据输入到达时间和最大深度给出延迟约束,使用SAT公式以在延迟约束下找到最小面积的网络。该算法在工艺映射器上实现,其运行时间与切割的数量成正比,因此适用于大型网络。结果表明,可以有效解决6输入函数问题。文献^[19]通过扩展SAT公式,除了针对布尔逻辑网络的大小或深度等传统成本指标进行优化外,还考虑了需要同时考虑的许多复杂约束的优化应用。文章首先通过比较用于精确综合的不同SAT编码,考虑延迟约束,找到针对大小或深度的最佳优化。其次考虑了复杂约束问题的精确综合,并解决了由于不同原语引起的约束和由于必须执行综合的逻辑表示的限制。

上述精确综合方法不能高效地为具有大于10个变量的函数找到紧凑的逻辑网络,Riener等人^[20]改进了精确综合的可扩展性,提出了一种以 k -LUT作为布尔逻辑网络节点的重写算法,适用性更加广泛。作者使用支持布尔无关项(don't-care)的基于SAT的精确综合引擎来即时计算替换项,并将所有可能的替换候选存储在冲突图中,从中计算出可兼容替换网络的最大子集。在此基础上,Riener等人^[21]设计了一个通用的再综合(resynthesis)框架,用于优化用多级逻辑表示、切割计算算法和再综合算法参数化的布尔逻辑网络。该框架基于精确DAG感知重写引擎,以即插即用的方式实现强大的优化算法。使用不同的再综合算法,包括递归的自上而下不相交支持分解、精确综合和数据库查找表的算法,可以应用于不同的多级逻辑表示,例如AIG, MIG以及LUT网络等,大大增加了算法的使用场景。

4.4 面向新兴计算范式的精确综合

新兴计算范式采用新型逻辑原语和计算策略,如采用蕴涵、多数逻辑门、阈值门(threshold logic gate)等,近似计算则放宽了对逻辑函数的精度要求,加入了一个新的维度优化PPA。因此,面向新兴计算范式的综合优化面临更多的优化约束,且缺乏可参考的设计。而精确综合可以有效对约束进行SAT编码并求解。

近年来,可逆逻辑电路的综合逐渐受到重视,同时还有其他基于函数转换或更高级别的函数表示

的方法,如异或乘积和(Exclusive-or Sum-Of-Products, ESOP),它们能够处理更大的电路,虽然其特点保证无需其他最优性或接近最优性的约束,但找到具有最少乘积项的ESOP形式是精确综合的目标。基于SAT的精确综合常被运用在可逆电路或高级函数表示的综合方法中。对于实现所需功能的电路,精确综合算法确定给定功能的最小电路实现,即分别具有最少门数或成本的电路。

Große等人^[22]在具有正控制线的Toffoli门电路中,分析SAT以及Toffoli门的特点,设计了适用于Toffoli门的SAT编码。设计的迭代算法将Toffoli网络综合问题表述为一系列决策问题,可以找到更小的给定函数,该选择编码方式对于生成最优网络至关重要。Wille等人^[23]则在带有负控制线的Toffoli门中,实现了SAT编码。考虑负控制线会导致电路比目前最优逻辑门数量的实现更小,所组成的可逆电路的精确综合效率更高。Kole等人^[24]在此基础上,实现了3元可逆逻辑的基于SAT的精确综合。作者将表示可逆3元真值表的置换作为输入,将由实现置换的广义3元Toffoli门组成的可逆电路作为输出。该方法提供了一个基准,可以与其他非精确综合的性能进行比较。

Riener等人^[25]通过将ESOP综合问题编码为SAT问题,提出一种使用SAT计算ESOP形式的精确综合方法,为给定的布尔函数找到1个或多个ESOP形式。类似于自底向上的基于SAT的精确综合,作者从可能不完全指定的布尔函数形式的规范开始,迭代地构造一个布尔约束满足问题,当且仅当具有实现规范的 k (最初 $k=1$)乘积项的ESOP形式存在时,该问题是可满足的,并返回具有 k 个乘积项的ESOP形式。这种基于SAT的精确综合方法不依赖应用于一对乘积项的立方变换,而是能够优化乘积项的小子集,有望成为新一代启发式ESOP优化方法的支柱。

随机计算(Stochastic Computing, SC)使用转换为随机比特流的二进制数实现计算复杂的算术,SC的容错特性使其被应用在大量应用中。然而随机电路综合允许比经典逻辑综合更大的解决方案空间,以往的启发式方法只探索有限的解空间子集,无法保证结果的最优性,因此基于精确综合的方法被大量运用在SC中。

Wang等人^[26]提出了一种基于输入误差边界进行搜索的方法来寻找目标函数的最佳逼近多项式,保证了最佳逼近多项式的最优性。为了优化电路的大小,作者在精确综合过程中考虑逻辑门面积的方法,并设计了基于SAT的精确综合方法,可以直接

综合面积最优的SC电路，无需工艺映射，同时提出了部分赋值(Partial Of Assignment, POA) 和多粒度搜索(Multi Granularity Search, MGS)来减少SAT求解器的搜索空间。He等人^[27]以MIG作为基本逻辑表示，使用基于SAT的精确综合来综合最佳SC电路。作者使用立方体赋值(cube assignment)分解问题向量，以解决可扩展性问题，加速精确综合。同时，为了探索任意大小的通用电路，作者设计了一种基于精确综合和启发式的混合算法，其中精确综合被用作决策者来减少解树中的分支数量，从而得到更小的面积延迟积。

作为一种内存计算的方法，忆阻电路允许数据存储和逻辑运算。Chen等人^[28]提出了一种精确综合方法来推导蕴涵(IMPLication, IMPLY)逻辑网络，对所有3输入布尔函数进行综合，无需中间步骤即可直接获得最优的 IMPLY 逻辑网络，较基于AIG的综合算法比有优化。

5 展望

由于精确综合的高计算复杂度，以上所提出的精确综合算法对小函数优化具有较高的效率，但它不适用于超大规模函数，直接对大规模逻辑网络进行精确综合寻优面临困难。SAT编码的优势在于它可以灵活地处理不同的成本函数，并且可以扩展到不同的场景。因此，将基于SAT 的精确综合算法应用于大型函数的小型子网络，可以保证大规模网络的局部最优性。本节简要阐述了目前精确综合所面临的困境和挑战，并尝试对未来可能对精确综合产生重大影响的高新技术进行了讨论，同时对未来的发展趋势进行预测。

5.1 SAT编码

首先，选择合适的SAT编码方法和SAT求解器的不同启发式算法会在一定程度上优化精确综合运行时间。文献^[1]表明，选择编码和对称性破拆的正确组合会极大地影响综合运行的时间，并且选择合适的SAT求解器启发式算法同样会加速精确综合的运行时间。因此一个研究方向是开发专门解决精确综合问题的特定领域SAT求解器，文献^[29]考虑到精确解可以帮助精确综合求解加速，设计了一种基于半张量积的SAT求解器，通过矩阵数理计算替换传统布尔逻辑运算的新型求解方法，得到SAT问题的全部精确解，可以用于筛选精确综合的最优解，提高求解的质量和速度。文献^[30]中，一种基于电路信息的SAT求解器已经成功应用于逻辑综合算法。由于该求解器可以事先知道问题逻辑网络的结构，其具有更快的约束传播和可变的启发式方法。

5.2 DAG拓扑结构

其次，选择合适的DAG拓扑结构可以显著减少精确综合运行时间。文献^[3]中，一个未解决的问题是需要搜索或生成“好”的DAG拓扑。例如，某些DAG可以表达更多信息或某些拓扑比其他拓扑可以表示更多的功能，因此能否找到一些特征来表征更具表达性的特定DAG拓扑。未来的工作中可以根据不同的策略找到枚举拓扑结构，以实现进一步的加速，或构建条件概率分布，在给定函数下，可以使用哪些类型的拓扑来表示该函数。此外，只需要检查包含潜在有效拓扑的空间部分，利用DAG拓扑即等于划分综合搜索空间。另一个研究方向是使用基于图的拓扑作为基于SAT的精确综合算法的基础，作为其并行性的来源。

5.3 人工智能

最后，利用人工智能(Artificial Intelligence, AI)技术优化EDA中的算法受到广泛关注^[31]。传统启发式算法通常需要对每个待求解问题进行初始化，在面对逻辑综合优化问题时需要大量的时间和资源，缺乏知识积累。相反，机器学习可以用于自动学习新的逻辑综合算法或改进算法本身，避免重复的复杂分析。因此，基于AI的逻辑综合方法是解决这一问题的有效途径之一。

许多逻辑综合优化问题可以归结为组合优化问题。基于AI的方法，一方面可以利用AI推理确定综合流(synthesis flow)，例如，逻辑综合引擎反复对子电路进行平衡(balance)、重写(rewrite)、重构(refactor)和重替换(re-substitution)等变换，以提高结果的质量。另一方面对逻辑网络进行自动划分，智能推断所采用的综合策略。在精确综合方面，AI可以对如何划分小规模电路进行学习，提高子电路的质量；同样也可在给定不同逻辑原语集合的情况下，对划分后的子电路进行精确综合，并有效评估综合的效果，确定是否为最优网络。

6 结束语

精确综合是一种通用的逻辑综合技术，可应用于逻辑优化、工艺映射、新兴技术的综合等。对于传统精确综合，存在运行时间无法预测，不足以有效地为具有大于6个变量的函数找到面积或深度最优的逻辑网络等问题。与传统手段相比，近年来，随着SAT约束求解能力的进步，结合SAT约束求解的精确综合确定为进一步发展的方向，其性能也得到了大规模提升。同时，基于SAT的精确综合可以集成任意约束，实现在任意条件下的高效精确综合，为新兴纳米电路物理条件约束下的综合提供全新思路。

参 考 文 献

- [1] HAASWIJK W J. SAT-based exact synthesis for multi-level logic networks[D]. EPFL, 2019. doi: [10.5075/epfl-thesis-9404](https://doi.org/10.5075/epfl-thesis-9404).
- [2] KNUTH D E. The Art of Computer Programming[M]. Upper Saddle River: Addison-Wesley Professional, 2005.
- [3] HAASWIJK W, SOEKEN M, MISHCHENKO A, *et al.* SAT-based exact synthesis: Encodings, topology families, and parallelism[J]. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2020, 39(4): 871–884. doi: [10.1109/TCAD.2019.2897703](https://doi.org/10.1109/TCAD.2019.2897703).
- [4] QUINE W V. The problem of simplifying truth functions[J]. *The American Mathematical Monthly*, 1952, 59(8): 521–531. doi: [10.1080/00029890.1952.11988183](https://doi.org/10.1080/00029890.1952.11988183).
- [5] ERNST E A. Optimal combinational multi-level logic synthesis[D]. [Ph. D. dissertation], The University of Michigan, 2009.
- [6] EÉN N. Practical SAT-a tutorial on applied satisfiability solving[R]. 2007.
- [7] SOEKEN M, AMARÙ L, GAILLARDON P E, *et al.* Exact synthesis of majority-inverter graphs and its applications[J]. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2017, 36(11): 1842–1855. doi: [10.1109/TCAD.2017.2664059](https://doi.org/10.1109/TCAD.2017.2664059).
- [8] CHU Zhufei, SOEKEN M, XIA Yinshui, *et al.* Advanced functional decomposition using majority and its applications[J]. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2020, 39(8): 1621–1634. doi: [10.1109/TCAD.2019.2925392](https://doi.org/10.1109/TCAD.2019.2925392).
- [9] HAASWIJK W, SOEKEN M, MISHCHENKO A, *et al.* SAT based exact synthesis using DAG topology families[C]. The 55Th ACM/ESDA/IEEE Design Automation Conference (DAC), San Francisco, USA, 2018: 1–6. doi: [10.1109/DAC.2018.8465888](https://doi.org/10.1109/DAC.2018.8465888).
- [10] HAASWIJK W J, SOEKEN M, AMARU L, *et al.* LUT mapping and optimization for majority-inverter graphs[C]. The 25th International Workshop on Logic & Synthesis (IWLS), Austin, USA, 2016.
- [11] HAASWIJK W, SOEKEN M, AMARÙ L, *et al.* A novel basis for logic rewriting[C]. 2017 22nd Asia and South Pacific Design Automation Conference (ASP-DAC), Chiba, Japan, 2017: 151–156. doi: [10.1109/ASPDAC.2017.7858312](https://doi.org/10.1109/ASPDAC.2017.7858312).
- [12] CHU Zhufei, HAASWIJK W, SOEKEN M, *et al.* Exact synthesis of Boolean functions in majority-of-five forms[C]. 2019 IEEE International Symposium on Circuits and Systems (ISCAS), Sapporo, Japan, 2019: 1–5. doi: [10.1109/ISCAS.2019.8702141](https://doi.org/10.1109/ISCAS.2019.8702141).
- [13] MARAKKALAGE D S, TESTA E, RIENER H, *et al.* Three-input gates for logic synthesis[J]. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2021, 40(10): 2184–2188. doi: [10.1109/TCAD.2020.3032625](https://doi.org/10.1109/TCAD.2020.3032625).
- [14] HAASWIJK W, TESTA E, SOEKEN M, *et al.* Classifying functions with exact synthesis[C]. 2017 IEEE 47th International Symposium on Multiple-Valued Logic (ISMVL), Novi Sad, Serbia, 2017: 272–277. doi: [10.1109/ISMVL.2017.44](https://doi.org/10.1109/ISMVL.2017.44).
- [15] PETKOVSKA A, SOEKEN M, DE MICHELI G, *et al.* Fast hierarchical NPN classification[C]. 2016 26th International Conference on Field Programmable Logic and Applications (FPL), Lausanne, Switzerland, 2016: 1–4. doi: [10.1109/FPL.2016.7577306](https://doi.org/10.1109/FPL.2016.7577306).
- [16] HUANG Zheng, WANG Lingli, NASIKOVSKIY Y, *et al.* Fast Boolean matching based on NPN classification[C]. 2013 International Conference on Field-Programmable Technology (FPT), Kyoto, Japan, 2013: 310–313. doi: [10.1109/FPT.2013.6718374](https://doi.org/10.1109/FPT.2013.6718374).
- [17] ZHOU Xuegong, WANG Lingli, and MISHCHENKO A. Fast exact NPN classification by Co-designing canonical form and its computation algorithm[J]. *IEEE Transactions on Computers*, 2020, 69(9): 1293–1307. doi: [10.1109/TC.2020.2971466](https://doi.org/10.1109/TC.2020.2971466).
- [18] SOEKEN M, DE MICHELI G, and MISHCHENKO A. Busy man's synthesis: Combinational delay optimization with SAT[C]. 2017 Design, Automation & Test in Europe Conference & Exhibition (DATE), Lausanne, Switzerland, 2017: 830–835. doi: [10.23919/DATE.2017.7927103](https://doi.org/10.23919/DATE.2017.7927103).
- [19] SOEKEN M, HAASWIJK W, TESTA E, *et al.* Practical exact synthesis[C]. 2018 Design, Automation & Test in Europe Conference & Exhibition (DATE), Dresden, Germany, 2018: 309–314. doi: [10.23919/DATE.2018.8342027](https://doi.org/10.23919/DATE.2018.8342027).
- [20] RIENER H, HAASWIJK W, MISHCHENKO A, *et al.* On-the-fly and DAG-aware: Rewriting Boolean networks with exact synthesis[C]. 2019 Design, Automation & Test in Europe Conference & Exhibition (DATE), Florence, Italy, 2019: 1649–1654. doi: [10.23919/DATE.2019.8715185](https://doi.org/10.23919/DATE.2019.8715185).
- [21] RIENER H, MISHCHENKO A, and SOEKEN M. Exact DAG-aware rewriting[C]. 2020 Design, Automation & Test in Europe Conference & Exhibition (DATE), Grenoble, France, 2020: 732–737. doi: [10.23919/DATE48585.2020.9116379](https://doi.org/10.23919/DATE48585.2020.9116379).
- [22] GROBE D, WILLE R, DUECK G W, *et al.* Exact multiple-control Toffoli network synthesis with SAT techniques[J]. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2009, 28(5): 703–715. doi: [10.1109/TCAD.2009.2017215](https://doi.org/10.1109/TCAD.2009.2017215).
- [23] WILLE R, SOEKEN M, PRZIGODA N, *et al.* Exact

- synthesis of Toffoli gate circuits with negative control lines[C]. The 2012 IEEE 42nd International Symposium on Multiple-Valued Logic, Victoria, Canada, 2012: 69–74. doi: [10.1109/ISMVL.2012.71](https://doi.org/10.1109/ISMVL.2012.71).
- [24] KOLE A, RANI P M N, DATTA K, *et al.* Exact synthesis of ternary reversible functions using ternary Toffoli gates[C]. 2017 IEEE 47th International Symposium on Multiple-Valued Logic (ISMVL), Novi Sad, Serbia, 2017: 179–184. doi: [10.1109/ISMVL.2017.51](https://doi.org/10.1109/ISMVL.2017.51).
- [25] RIENER H, EHLERS R, DE O SCHMITT B, *et al.* Exact synthesis of ESOP forms[M]. DRECHSLER R and SOEKEN M. Advanced Boolean Techniques: Selected Papers from the 13th International Workshop on Boolean Problems. Cham: Springer, 2020: 177–194. doi: [10.1007/978-3-030-20323-8_8](https://doi.org/10.1007/978-3-030-20323-8_8).
- [26] WANG Xuan, CHU Zhufei, and QIAN Weikang. MinSC: An exact synthesis-based method for minimal-area stochastic circuits under relaxed error bound[C]. 2021 IEEE/ACM International Conference On Computer Aided Design (ICCAD), Munich, Germany, 2021: 1–9. doi: [10.1109/ICCAD51958.2021.9643580](https://doi.org/10.1109/ICCAD51958.2021.9643580).
- [27] HE Xiang and CHU Zhufei. Stochastic circuit synthesis via satisfiability[J]. *Integration*, 2022, 84: 84–91. doi: [10.1016/J.VLSI.2021.11.003](https://doi.org/10.1016/J.VLSI.2021.11.003).
- [28] CHEN Lin and CHU Zhufei. Towards optimal logic representations for implication-based memristive circuits[C]. 2020 China Semiconductor Technology International Conference (CSTIC), Shanghai, China, 2020: 1–3. doi: [10.1109/CSTIC49141.2020.9282401](https://doi.org/10.1109/CSTIC49141.2020.9282401).
- [29] PAN Hongyang and CHU Zhufei. A semi-tensor product based SAT all solutions solver[C]. The 2021 CCF Integrated Circuit Design and Automation Conference (CCFDAC), Wuhan, China, 2020.
- [30] ZHANG Heteng, JIANG J H R, and MISHCHENKO A. A circuit-based SAT solver for logic synthesis[C]. 2021 IEEE/ACM International Conference On Computer Aided Design (ICCAD), Munich, Germany, 2021: 1–6. doi: [10.1109/ICCAD51958.2021.9643505](https://doi.org/10.1109/ICCAD51958.2021.9643505).
- [31] HOSNY A, HASHEMI S, SHALAN M, *et al.* DRiLLS: Deep reinforcement learning for logic synthesis[C]. 2020 25th Asia and South Pacific Design Automation Conference (ASP-DAC), Beijing, China, 2020: 581–586. doi: [10.1109/ASP-DAC47756.2020.9045559](https://doi.org/10.1109/ASP-DAC47756.2020.9045559).
- 储著飞：男，副教授，博士生导师，研究方向为逻辑综合与优化。
潘鸿洋：男，硕士生，研究方向为逻辑综合与优化。
- 责任编辑：余 蓉