

二叉决策图映射电路的面积和延时优化

张会红 陈治文 汪鹏君*

(宁波大学电路与系统研究所 宁波 315000)

摘要: 二叉决策图(BDD)是一种数据结构, 广泛应用于数字电路的逻辑综合、测试和验证等领域。将BDD每个结点映射成2选1数据选择器(MUX)可得到BDD映射电路。该文提出一种BDD映射电路的面积和延时优化方法。首先把待优化电路转换成BDD形式, 然后逐一搜索BDD中存在的菱形结构, 进而通过路径优化实现结点的删减和控制变量的更改, 并将所得结果BDD映射成MUX电路, 最后用多个MCNC基准电路进行测试, 将该文方法与经典综合工具BDS, SIS等方法相比较, BDD总结点数比BDS减少了55.8%, 映射电路的面积和延时比SIS分别减小了39.3%和44.4%。

关键词: 电路优化; 二叉决策图; 数据选择器; 延时优化

中图分类号: TN79+1; TP391.7

文献标识码: A

文章编号: 1009-5896(2019)03-0725-07

DOI: 10.11999/JEIT180443

Area and Delay Optimization of Binary Decision Diagrams Mapped Circuit

ZHANG Huihong CHEN Zhiwen WANG Pengjun

(Institute of Circuits and Systems, Ningbo University, Ningbo 315000, China)

Abstract: Binary Decision Diagrams (BDD) is a data structure that can be used to describe a digital circuit. By replacing each node in a BDD with a 2-to-1 Multiplexer (MUX), a BDD can be mapped to a digital circuit. An area and delay optimization method on BDD mapped circuit is presented. A traditional Boolean circuit is converted into BDD form, and then diamond structure constructed by nodes is searched in the BDD, corresponding nodes are deleted and control signals of the modified nodes are updated by paths optimization, finally, the result BDD is mapped to a MUX circuit. The proposed method is test by a number of Microelectronics Center of North Carolina (MCNC) Benchmarks. Compared with the classical synthesis tools Sequential Interactive System (SIS) and BDD-based logic optimization System (BDS), the average number of nodes by the proposed methods is 55.8% less than that of BDS, and average circuit's area and delay are reduced by 39.3% and 44.4% than that of the SIS, respectively.

Key words: Circuit optimization; Binary Decision Diagrams (BDD); Multiplexer; Delay optimization

1 引言

面积和延时一直是集成电路设计性能的重要指标^[1]。传统的面积和延时优化主要基于与、或、非运算的传统布尔逻辑(Traditional Boolean logic, TB logic)。然而, TB逻辑并不是所有逻辑电路的

最佳实现方式。针对各种复杂逻辑结构的电路, 已有多逻辑表现形式应用在电路优化中: Reed-Muller(RM)逻辑^[2,3]在异或和与或密集型电路上应用广泛; 与非门驱动的AIG(And-Inverter Graph)逻辑^[4,5]和 Majority门驱动的MIG(Majority-Inverter Graph)逻辑^[6]在电路优化中也取得了不错的效果。研究发现, BDD(Binary Decision Diagrams)的优化可用于电路逻辑综合^[7]、自动化检测与验证技术^[8]、可满足性问题^[9]等领域。

文献^[10,11]以BDD电路的具体性能为目标, 将启发式智能算法用于优化BDD结点和路径; 文献^[12]在逻辑综合工具BDS(BDD-based logic optimization System)中设计了BDDlopt算法, 对BDD结点执行分解后再综合, 在AND/OR和XOR密集型

收稿日期: 2018-05-10; 改回日期: 2018-11-21; 网络出版: 2018-12-04

*通信作者: 汪鹏君 wangpengjun@nbu.edu.cn

基金项目: 国家自然科学基金(61474068, 61306041), 浙江省公益技术应用研究计划项目(2016C31078), 宁波大学研究生科研创新基金 Foundation Items: The National Natural Science Foundation of China (61474068, 61306041), Zhejiang Province Public Welfare Technology Application Research Project (2016C31078), The Scientific Research Foundation of Graduate School of Ningbo University

电路的面积上取得了良好的优化结果；文献[13]提出了SMTBDD(Shared Multi Terminal Binary Decision Diagram)，该BDD进行多重切割时不需要更改变量集，切割后的电路所需要的LUT较少；文献[14]提出DDBDD(Delay-Driven BDD)方法，该方法在BDD分解时采用线性扩展技术，并应用动态规划算法高效地搜索和优化局部BDD的冗余结构，大大减小延时；文献[7]提出BBDD(Biconditional BDD)方法，该方法生成的部分结点控制变量为两变量的异或/与或，对于XOR密集型电路的优化效果明显。然而，基于智能算法的BDD优化存在最优解问题；BDS在执行功能分解时，往往以面积为代价得到更小的延时；SMTBDD和DDBDD方法中，映射的LUT随着输入变量增多，其面积消耗剧增，延时优化的代价相当明显；BBDD方法适用于XOR密集型电路，其结点控制变量是两变量的组合，变量排序极其复杂，仅适用于输入较少的电路。

本文针对以上问题提出基于结点菱形结构的优化算法，首先应用CUDD^[15]获得电路的最优变量序下的BDD，然后在BDD中搜索可能存在的菱形结构，根据结构的类型删减结点或更改控制变量，最后将结果BDD映射成电路。该算法通过优化BDD路径以删除或合并冗余结点，实现电路BDD及其映射电路的面积优化，且当冗余结点位于关键路径时，可实现面积和延时同时优化；BDD结点和路径的减少，还有利于BDD相关的自动化检测和验证技术的发展。

2 函数的BDD

对于任意一个布尔表达式： $f = B^n \rightarrow B$ ，由香农展开式可得： $f = \bar{x}f_0 + xf_1$ ，其中， f_0 表示控制变量 $x=0$ 时 f 的取值； f_1 表示 $x=1$ 时 f 的表达式。 f 对于任意变量 x_i 的香农展开式如式(1)所示

$$\begin{aligned} f(x_1, x_2, \dots, x_i, \dots, x_n) \\ = \bar{x}_i f(x_1, x_2, \dots, x_{i-1}, 0, x_{i+1}, \dots, x_n) \\ + x_i f(x_1, x_2, \dots, x_{i-1}, 1, x_{i+1}, \dots, x_n), \\ i = 1, 2, \dots, n \end{aligned} \quad (1)$$

n 变量布尔函数的BDD可通过对每个变量按照式(1)所示的香农展开得到。BDD是一种具有终结点和非终结点的有向无环图(directed acyclic graph)，非终结点的两条扇出边由输入变量决定，终结点直接取值为逻辑值0或1。考虑式(2)的表达式

$$f = \bar{a}bc + \bar{a}b\bar{c} + ab\bar{c} + a\bar{b}\bar{c} \quad (2)$$

式(2)的BDD如图1(a)所示。该BDD中最后一行的矩形结点均为终结点，即函数 f 在具体路径下

的最终取值；圆形结点为非终结点，每个非终结点有两个扇出边，分别是当前变量取0, 1时的结果， a, b, c 表示所在水平线上所有结点的控制变量，简称变量。为方便讨论，本文约定：当前结点的变量表示为 $\text{var}(f)$ ；扇出边表示为 edge ，扇出边指向结点为 $\text{edge}(f)$ ；变量为0/1时的扇出边表示为 low/high ，相应边指向结点为 $\text{low}(f)/\text{high}(f)$ 。未化简的BDD包含 2^n-1 个非终结点，随着输入变量 n 的增加，结点数目呈指数增长，直接映射所得电路的面积也将同样急剧增加，因此电路映射之前需要进行BDD化简。对于图1(a)所示BDD，在指定变量顺序下使用ITE算子^[16]直接构建化简的有序BDD如图1(b)所示，其中BDD变量序为： $a \rightarrow b \rightarrow c$ 。这样的BDD称为化简后的有序BDD(Reduced Ordered BDD, ROBDD)，文中未作说明时均视为ROBDD。

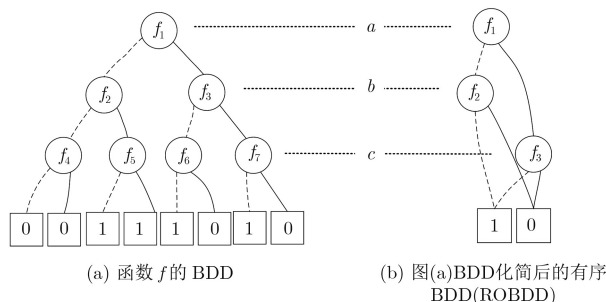


图1 式(1)函数 f 的BDD及其化简后的ROBDD

3 BDD映射电路及优化

任何一个布尔表达式，都可以使用SIS (Sequential Interactive System)^[17]工具将其转换成标准的BLIF(Berkeley Logic Interchange Format)电路文件，这也是CUDD的输入文件格式。使用CUDD^[15]工具可构造电路的BDD，BDD经优化后映射成电路，其中结点映射成MUX，结点的变量映射成MUX的控制变量。文献[10]应用该方法设计了基于BDD的低功耗电路，证明了BDD映射电路的适用性、有效性。

3.1 菱形结构

未优化的BDD可能含有大量的冗余结点，直接映射成电路时会造成电路面积开销大、延时不理想。BDS等综合工具针对全局结点进行门的分解，大幅度优化了面积和延时。但是在布尔分解时的结点重复利用导致了大面积换取小延时的局面。因此，本文充分利用BDD中存在的菱形结构进行结点分解，取得了同时优化电路面积和延时的效果。

以下给出定义便于进一步分析，其中的根结点可以是全局的或局部的。

定义 对于根结点 F 的BDD，结点 $\text{low}(F)$ 和结

点 $\text{high}(F)$ 的扇入度均为1时, 若 $\text{edge}[\text{low}(F)] = \text{edge}[\text{high}(F)]$, 且 $\text{var}[\text{low}(F)] = \text{var}[\text{high}(F)]$ 时, 则结点 $\{F, \text{low}(F), \text{high}(F), \text{edge}[\text{low}(F)]\}$ 和相应边构成的形如“ $\diamond$ ”的结构称为菱形结构, 记“ $\diamond_F$ ”。

如图2, 结点 f_2 和 f_3 的变量相同, 且 $\text{high}(f_2) = \text{low}(f_3)$, 则边 $\{e_1, e_2, e_3, e_4\}$ 和结点 $\{f_1, f_2, f_3, f_5\}$ 相连可得一个菱形结构, 如图2中阴影所示。本文的优化在菱形结构和其必要输出结点上进行, 图2中, 结点 $\{f_1, f_2, f_3, f_5\}$ 构成菱形结构; 结点 $\{f_4, f_6\}$ 为必要输出结点。

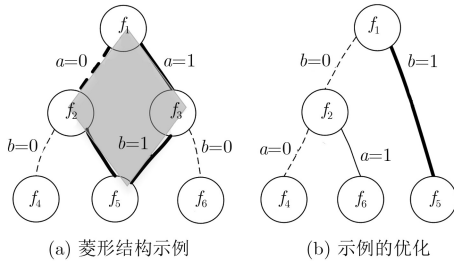


图2 菱形结构及其优化

从二叉树的角度考虑图2(a)中 $\diamond_{f_1}$ 的优化, 根据满二叉树的定理^[8]: 对于任意的满二叉树, 其叶子结点数 n_O 与非叶子结点数 n_I 满足 $n_O = n_I + 1$, 即一个结点数目为 N 的BDD最多有 $(N+1)/2$ 个输出。未优化图中有3个输入结点(非叶子结点)和3个输出结点(叶子结点), 由满二叉树定理可知输出个数为3时, 最少仅需要2个输入结点。

针对含菱形结构的BDD, 本文提出基于路径优化的BDD重构优化。考虑一个含菱形结构的BDD, 首先找出菱形结构内由根结点到输出结点的2条路径, 然后合并该路径为1条路径, 最后重构根结点到其他输出结点的路径。重构时不会增加额外的路径, 因此总路径的减少保证了总结点数的减少。以图2(a)为例, 从根结点到输出结点一共有4条路径。其中由菱形结构覆盖的2条路径($\bar{a}b$ 和 ab)可以合并成1条路径(b)。通过重构其他路径, 最终实现由图2(a)到图2(b)所示的BDD重构优化。其他情况下菱形结构覆盖的路径合并如图3所示, 框内为合并后的路径。

据此, 对于含有菱形结构的局部BDD, 应用

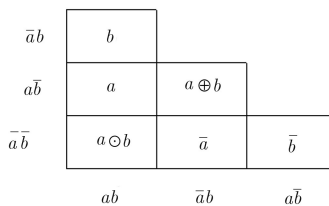


图3 路径合并

以下方法进行优化:

步骤1 提取出含有菱形结构的局部BDD, 该BDD包含菱形结构和相应的输出结点;

步骤2 统计BDD中的所有路径, 以及各路径达到的输出结点, 记为复合路径, 形如 xyf 。其中, x 和 y 表示变量, f 表示输出结点;

步骤3 将输出结点相同的复合路径合为一组, 将该组的复合路径合并成新的复合路径 η ;

步骤4 若复合路径未处理完毕, 执行步骤5; 否则, 执行步骤6;

步骤5 将输出结点不同的复合路径合为一组, 删除该组复合路径中逻辑相反的变量直到剩下唯一一组逻辑相反的变量, 所得路径为2条新的复合路径 γ_1 和 γ_2 ;

步骤6 根据所有的复合路径 η 和 γ (若存在)重构局部BDD。

例如某BDD如图2(a), 统计所有的复合路径为: $\bar{a}bf_4$, $\bar{a}bf_5$, abf_5 和 abf_6 。 $\bar{a}bf_5$ 和 abf_5 的输出结点相同, 因此合并后可得复合路径 η 为 bf_5 。 $\bar{a}bf_4$ 和 abf_6 的输出结点不同, 其中变量 a 以原变量和反变量形式出现, 无其他相反的变量, 故复合路径 γ_1 , γ_2 为 $\bar{a}bf_4$, abf_6 。根据复合路径 η 和 γ 重构的BDD如图2(b)。

含菱形结构的BDD均可应用以上步骤进行优化, 路径的减少保证了优化效果, 优化的BDD均满足 $n_O = n_I + 1$ 。此外, 当菱形结构中某些条件改变时也能应用上述步骤进行优化。目前已发现的条件有: (1)根结点的两个扇出结点的变量不相同; (2)根结点的1条路径直接指向输出结点。相应的优化示例如图4、图5所示。

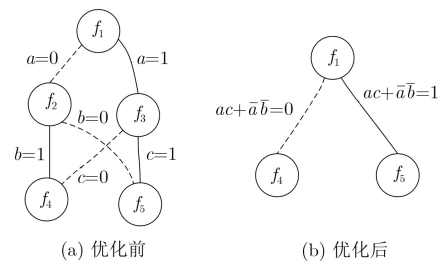


图4 优化示例

3.2 面积模型

本文对于BDD结构的优化建立在最优变量序的BDD基础上, 目前基于变量排序的BDD优化研究已经相对成熟, CUDD是该领域经典、高效的工具。本文应用CUDD构建最优变量序的BDD, 再通过优化其局部结构来实现面积的进一步优化。面积估算模型基于mcnc.genlib工艺库建立, 首先统

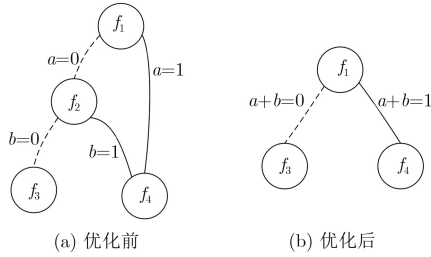


图5 优化示例

计优化后映射电路中MUX与XOR, OR与AND和反相器(INV)等单元门的数量, 并取反相器面积为1个单位, 电路总面积以各种单元门的加权之和表示, 具体如式(3)所示

$$S_{\text{area}} = 5n_{\text{mux/xor}} + 3n_{\text{or/and}} + n_{\text{inv}} \quad (3)$$

其中, $n_{\text{mux/xor}}$ 为MUX与XOR数目, $n_{\text{or/and}}$ 为OR与AND数目, n_{inv} 为反相器数目。

3.3 延时模型

数字电路的延时可以通过其级联情况来估计。采用单位延时模型估计时, 将电路中的多输入门分解成2输入门, 每个2输入门的延时为1个单位延时, 电路的延时可估算为关键路径上各结点传输时延之和, 考虑1个结点 f , 其结点延时用式(4)计算^[17]

$$t_f = 1 + \text{Max}(t_{f_0}, t_{f_1}, t_c) \quad (4)$$

其中, t_f 表示MUX结点 f 输出延时, t_{f_0} 和 t_{f_1} 为该结点两个输入的传输延时, t_c 表示 f 的控制变量的传输延时, 控制变量包括XOR, OR和AND结点, $t_c = 1 + \text{Max}(t_1, t_2)$, t_1 和 t_2 为该结点两个输入的传输延时。

3.4 BDD电路的优化

本文关于BDD电路的优化包括面积、延时优化, 提出的基于菱形结构的优化方法主要通过优化BDD路径, 使得结点减少, 以此减小电路面积。当关键路径经过待优化的局部BDD时, 有选择性地优化关键路径上的结点来实现面积优化的同时减小延时。

面积优化时, 首先应用CUDD得到最优变量序下的BDD, 然后搜索BDD中的菱形结构, 再根据结构的类型优化该结构中的局部BDD。同时, 考虑到面积的节省, 当输出结点为终结点“0”或

“1”时, 在接下来的电路映射中可以选择面积更小的单元门。如图6, 某一输出结点为终结点“1”时, 最终可直接映射成1个3输入的OR门, 面积大小等同于1个2输入的OR门加上1个反相器的面积。当所有结构优化完成后, 将结果BDD映射成电路, 根据面积模型估算最终电路面积。

延时优化和面积优化同时进行。搜索到所有的菱形结构后, 优先考虑删除关键路径上的冗余结点, 或者将关键路径上的结点修改成其他结点的控制信号。当所有结构优化完成后, 根据延时模型估算最终电路延时。

根据以上内容, 本文提出基于菱形结构的BDD优化算法, 实现了BDD电路的面积和延时优化, 具体算法的伪代码如表1。

4 实验数据及分析

所提算法用C语言实现, 在Ubuntu操作系统下, 通过gcc编译后运行, 程序运行环境为Intel Core i5-3210M, 2.50 GHz CPU, 4 GB RAM, 测试电路均为MCNC Benchmarks。

算法读入一个pla格式电路文件, 应用CUDD构建其有序的化简后的BDD, 对BDD中每个结点搜索以其为根结点的菱形结构并完成相应的优化步骤, 优化完成后, 根据相应模型估算映射电路的面积和延时。

为说明本方法的面积优化效果, BDD优化完成后的结果与BDS, DDBDD处理结果对比见表2。

“电路”列为MCNC Benchmarks电路文件名; “输入/输出”列为相应电路输入数目/输出数目; “原结点”代表待优化BDD结点数; “BDS”, “DDBDD”表示应用BDS工具、DDBDD工具的优化结果; “本文方法”表示应用本文提出方法的优化结果。其中, “结点”表示优化后BDD结点数; “时间(s)”表示CPU运行时间, “rate₁”, “rate₂”分别表示本方法优化后结点数比BDS, DDBDD方法改善的结点数百分比, 计算方法如式(5)

$$\text{rate}_i = (N_x - N_p) / N_x \times 100\% \quad (5)$$

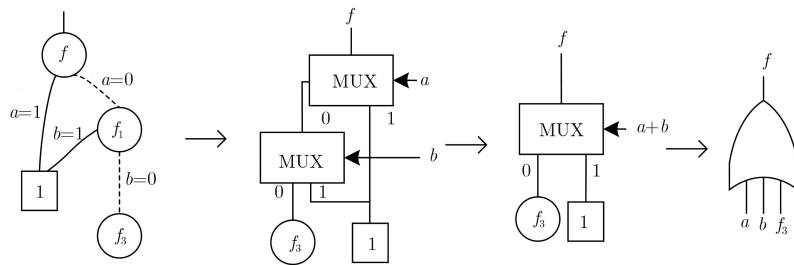


图6 优化示例

表 1 本文菱形结构BDD优化算法的伪代码

```

Begin algorithm
Read_pla_benchmark_circuit();
Apply_CUDD_package(); //obtain circuit's BDD
Number_nodes(); //number nodes from 1 to num_of_BDD
M=0;
While ( $M < \text{num\_of\_BDD}$ ) //handle all the non-terminals
    M=M+1;
    if(is_nodeM_diamond_structure()) //judge nodeM
        compute_mixed_paths();
        reconstruct_BDD();
    End if
End while
Output_logic_circuit();
Apply_area_model(); //calculate area and delay according to
model
Apply_delay_model();
Output_results();
Free_memory_unit();
End algorithm

```

其中, $i=1, 2$; N_x 表示采用BDS或DDBDD方法优化得到的结点数目; N_p 表示采用本方法优化得到的结点数目。

从表2可以看出,所提方法的优化结点数目明显少于BDS和DDBDD优化结果。BDS在进行布尔分解时会拆分原BDD中的共享结点,因此在分解了几乎所有MUX结点的同时,额外增加了大量

AND, OR等结点, 电路“table3”和“table5”很明显地体现了这一点; DDBDD将BDD映射成LUT, 其中大量使用了输入变量大于2的LUT, 即LUT k ($k>2$)。当 $k>2$ 时, 实际电路中MUX的数量是 2^k-1 个, 是指数增加的; 而所提方法在优化中, 复合路径的合并保证了结点数目的减少。

本文算法的面积和延时优化具体情况与其他方法的比较见表3。前2列含义同表2, “SIS”, “BBDD” 分别代表应用SIS, BBDD和本文方法的优化结果, “CGMP” 是文献[18]中提出的基于逻辑复合门的映射方法, 该方法通过对BDD分析搜索构造电路的逻辑复合门, 应用等效变换法选择最佳复合门完成电路优化。其中, “面积” 表示映射后的电路面积, “延时” 表示映射后的电路延时。 “ $\text{rate}_a/\text{rate}_d$ ” 表示本方法所得结果比前3种方法的面积/延时减小的百分比, 计算方法同 “ rate_1 ” 和 “ rate_2 ”, 把式(5)中的结点数目替换为对应的面积或延时即可。

从表3可以看出, 所提方法对于电路面积优化效果十分明显, 较其他方法均有改善。整体面积优化情况: BBDD优于SIS, SIS优于DDBDD。BBDD对于电路“e64”的面积优化尤其明显, 因此平均面积较好。SIS对于面积和延时的同时优化控制得较好。DDBDD映射成多输入LUT也带来了大量结点, 这样做的好处是延时相对较小。本文方法优化BDD时不产生额外结点, 因此在大部分电路面积上均占优势。从延时上分析, 本方法对于大部分电路的优化均优于除DDBDD外的其他方法。

表 2 所提方法的结点优化情况

[illegible]

表 3 所提方法的面积(a.u.)和延时(a.u.)优化情况

电路	输入/输出	SIS ^[17]		BBDD ^[7]		CGMP ^[18]		本文方法		rate _a /rate _d (%)		
		面积	延时	面积	延时	面积	延时	面积	延时	SIS	BBDD	CGMP
cordic	23/2	194	11	225	14	164	16	162	16	16.5/-45.5	28.0/-14.3	1.2/0
9sym	9/1	528	14	90	7	176	7	125	7	76.3/50.0	-38.9/0	29.0/0
rd84	8/4	402	14	190	7	215	6	166	6	58.7/57.1	12.6/14.3	22.8/0
t2	16/13	645	17	890	13	620	10	483	12	25.1/29.4	45.7/7.7	22.1/-20.0
duke2	22/29	1598	20	4410	17	1836	16	1438	15	10.0/25.0	67.4/11.8	21.7/6.3
alu4	14/8	1614	13	3000	13	2986	8	2532	8	-56.9/38.5	15.6/38.5	15.2/0
misex3	14/14	3668	21	4480	12	2530	8	2030	6	44.7/71.4	54.7/50.0	19.8/25.0
bc0	26/11	4025	24	6250	20	2640	16	2191	15	45.6/37.2	64.9/25.0	17.0/6.3
e64	65/64	4235	16	640	64	1453	12	639	7	84.9/56.3	0.2/89.1	56.0/41.7
table5	17/15	5069	24	4660	15	3024	14	2863	11	43.5/54.2	38.6/26.7	5.3/21.4
table3	14/14	5230	23	4365	12	3876	8	3473	8	33.6/65.2	20.4/33.3	10.4/0
cps	24/108	5525	20	—	—	5013	17	3990	14	27.8/30.0	—	20.4/17.6
平均										34.2/39.1	28.1/25.6	20.1/8.2

与SIS方法相比改进的程度十分明显, 因为BDD构建时按照变量序, 最终生成的BDD深度一定不大于变量个数, 这直接决定了基于BDD方法的优化电路的延时几乎不会大于变量个数, 这是和SIS方法不同的。另外, 从CPU运行时间来看, 本方法与BBDD接近, 大部分电路能在1 s内完成优化, 运行效率较高。

5 结束语

本文所提基于菱形结构的BDD优化算法有效删除了冗余结点, 优化了电路结构, 将原BDD分解成更少的结点和控制变量, 实现了BDD映射电路的面积和延时的优化。传统分解方法在MUX结点减少的同时会带来大量的其他结点, 面积和延时性能不太理想。经大量基准电路进行测试, 发现本算法相对于其他算法更有效地改善了电路的面积和延时, 为BDD的优化提出了新思路。未来的研究主要考虑BDD映射电路的低功耗性能, 以及在BDD验证中的应用。

参 考 文 献

- [1] DAS A, DEBNATH A, and PRADHAN S. Area, power and temperature optimization during binary decision diagram based circuit synthesis[C]. *Devices for Integrated Circuit*, Kalyani, India, 2017: 778–782.
- [2] 符强, 汪鹏君, 王铭波, 等. 求解FPRM电路极性优化问题的改进多目标粒子群算法[J]. *计算机辅助设计与图形学学报*, 2018, 30(3): 540–548. doi: [10.3724/SP.J.1089.2018.16297](https://doi.org/10.3724/SP.J.1089.2018.16297).
- [3] 符强, 汪鹏君, 童楠, 等. 基于MODPSO算法的FPRM电路多约束极性优化方法[J]. *电子与信息学报*, 2017, 39(3): 717–723. doi: [10.11999/JEIT160458](https://doi.org/10.11999/JEIT160458).
- [4] YU Cunxi, CIESIELSKI M, and MISHCHENKO A. Fast algebraic rewriting based on and-inverter graphs[J]. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2017, 37(9): 1907–1911. doi: [10.1109/TCAD.2017.2772854](https://doi.org/10.1109/TCAD.2017.2772854).
- [5] NOUREDDINE M and ZARAKET F. Model checking software with first order logic specifications using AIG solvers[J]. *IEEE Transactions on Software Engineering*, 2016, 42(8): 741–763. doi: [10.1109/TSE.2016.2520468](https://doi.org/10.1109/TSE.2016.2520468).
- [6] CHUN Chechung, YUNG Chichen, CHUN Yaowang, et al. Majority logic circuits optimisation by node merging[C]. *Design Automation Conference*, Chiba, Japan, 2017: 714–719.
- [7] AMARU L. New Data Structures and Algorithms for Logic Synthesis and Verification[M]. Switzerland: Springer International Publishing, 2017: 17–19.
- [8] FRIED D, TABAJARA L M, and VARDI M Y. BDD-Based boolean functional synthesis[C]. *International Conference on Computer Aided Verification*, Toronto,

algorithm for polarity optimization of FPRM circuits[J]. *Journal of Computer-Aided Design & Computer Graphics*, 2018, 30(3): 540–548. doi: [10.3724/SP.J.1089.2018.16297](https://doi.org/10.3724/SP.J.1089.2018.16297).

[3] 符强, 汪鹏君, 童楠, 等. 基于MODPSO算法的FPRM电路多约束极性优化方法[J]. *电子与信息学报*, 2017, 39(3): 717–723. doi: [10.11999/JEIT160458](https://doi.org/10.11999/JEIT160458).

FU Qiang, WANG Pengjun, TONG Nan, et al. Multi-constrained polarity optimization of large-scale FPRM circuits based on multi-objective discrete particle swarm optimization[J]. *Journal of Electronics & Information Technology*, 2017, 39(3): 717–723. doi: [10.11999/JEIT160458](https://doi.org/10.11999/JEIT160458).

[4] YU Cunxi, CIESIELSKI M, and MISHCHENKO A. Fast algebraic rewriting based on and-inverter graphs[J]. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2017, 37(9): 1907–1911. doi: [10.1109/TCAD.2017.2772854](https://doi.org/10.1109/TCAD.2017.2772854).

[5] NOUREDDINE M and ZARAKET F. Model checking software with first order logic specifications using AIG solvers[J]. *IEEE Transactions on Software Engineering*, 2016, 42(8): 741–763. doi: [10.1109/TSE.2016.2520468](https://doi.org/10.1109/TSE.2016.2520468).

[6] CHUN Chechung, YUNG Chichen, CHUN Yaowang, et al. Majority logic circuits optimisation by node merging[C]. *Design Automation Conference*, Chiba, Japan, 2017: 714–719.

[7] AMARU L. New Data Structures and Algorithms for Logic Synthesis and Verification[M]. Switzerland: Springer International Publishing, 2017: 17–19.

[8] FRIED D, TABAJARA L M, and VARDI M Y. BDD-Based boolean functional synthesis[C]. *International Conference on Computer Aided Verification*, Toronto,

- Canada, 2016: 402–421.
- [9] MESKI A, PENZEK W, SZRETER M, *et al.* BDD-versus SAT-based bounded model checking for the existential fragment of linear temporal logic with knowledge: Algorithms and their performance[J]. *Autonomous Agents and Multi-Agent Systems*, 2014, 28(4): 558–604. doi: [10.1007/s10458-013-9232-2](https://doi.org/10.1007/s10458-013-9232-2).
- [10] KERTTU M, LINDGREN P, DRECHSLER R, *et al.* Low power optimization technique for BDD mapped finite state machines[C]. International Workshop on Logic & Synthesis, San Diego, USA, 2007: 143–148.
- [11] SHIRINZADEH S, SOEKEN M, and DRECHSLER R. Multi-objective BDD optimization with evolutionary algorithms[C]. Conference on Genetic & Evolutionary Computation, Madrid, Spain, 2015: 751–758.
- [12] YANG Congguang and CIESIELSKI M. BDS: A BDD-based logic optimization system[J]. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2002, 21(7): 866–876. doi: [10.1109/TCAD.2002.1013899](https://doi.org/10.1109/TCAD.2002.1013899).
- [13] KUBICA M and KANIA D. SMTBDD: New form of BDD for logic synthesis[J]. *International Journal of Electronics and Telecommunications*, 2016, 62(1): 33–41. doi: [10.1515/eletel-2016-0004](https://doi.org/10.1515/eletel-2016-0004).
- [14] CHENG Lei, CHEN Deming, and WONG M. Martin DDBDD: Delay-Driven BDD synthesis for FPGAs[J]. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2008, 27(7): 1203–1213. doi: [10.1109/TCAD.2008.923088](https://doi.org/10.1109/TCAD.2008.923088).
- [15] SOMENZI F. CUDD: CU Decision Diagram package release 3.0.0[OL]. <http://vlsi.colorado.edu/~fabio/CUDD>, 2017.
- [16] BRACE K S, RUDELL R L, and BRYANT R E. Efficient implementation of a BDD package[C]. IEEE Design Automation Conference, Orlando, USA, 1991, 40–45.
- [17] SENTOVICH E M, SINGH K J, LAVAGNO L, *et al.* SIS: A system for sequential circuit synthesis[OL]. <https://embedded.eecs.berkeley.edu/pubs/downloads/sis/index.htm>, 2017.
- [18] 岑旭梦, 王伦耀, 夏银水. 基于逻辑复合门映射的电路面积优化[J]. 宁波大学学报, 2016, 29(4): 38–43.
- CEN Xumeng, WANG Lunyao, and XIA Yinshui. Area optimization based on the complex logic gates mapping[J]. *Journal of Ningbo University (NSEE)*, 2016, 29(4): 38–43.
- 张会红: 女, 1976年生, 副教授, 研究方向为集成电路设计与优化、控制理论与应用.
- 陈治文: 男, 1993年生, 硕士, 研究方向为集成电路逻辑优化.
- 汪鹏君: 男, 1966年生, 教授, 博士生导师, 研究方向为低功耗、高信息密度集成电路和安全芯片设计及理论研究.