

基于 Majority 逻辑门映射的电路面积优化

吴达斌, 王伦耀*, 夏银水

(宁波大学 信息科学与工程学院, 浙江 宁波 315211)

摘要: 提出了一种采用三输入 Majority 逻辑门(M 门)和反相器实现电路逻辑功能的电路面积优化算法. 该算法首先用动态规划策略完成电路的 AIG(And-Inverter Graph)切割, 然后通过搜索适合 M 门覆盖的 AIG 子结构以及与门和 M 门之间的等效替代, 将 AIG 转换为适合 M 门映射的 MIG (Majority-Inverter Graph), 最后用 Ω 法则和 Ψ 法则对 MIG 进行简化, 得到节点数更少的 MIG, 从而实现基于 M 门映射的电路面积优化. 优化算法用 C 语言实现, 并用 MCNC 电路测试. 结果表明, 相比于逻辑综合工具 ABC 的结果, 用 M 门映射后电路面积可得到平均约 12%左右的优化.

关键词: Majority 逻辑门; 映射算法; 逻辑优化

中图分类号: TP331

文献标志码: A

文章编号: 1001-5132 (2019) 01-0026-06

随着电路集成度、单位面积功耗等越来越高, 基于硅基半导体工艺的 CMOS 集成电路面临越来越大的挑战, 因此新的纳米电子器件在集成电路领域的应用受到了广泛关注, 这些纳米器件包括硅纳米线^[1-2]、碳纳米管^[3]、石墨烯^[4-5]、电阻式随机存取存储器^[6](Resistive Random-Access Memory, RRAM)、量子元胞自动机^[7](Quantum-dot Cellular Automata, QCA)、有机场效应晶体管^[8](Organic Field Effect Transistors, OFET)和许多其他类型的纳米器件等等. 而这些器件的基本功能可抽象为一个多数(Majority)表决器^[9], 本文称之为 Majority 逻辑门(简称 M 门). 因此, 逻辑函数基于 M 门的实现是利用上述纳米器件实现集成电路的关键环节.

在逻辑函数基于 M 门实现的研究中, 文献[9]提出将逻辑函数的 AOIG (And-Or-Inverter Graph) 转换为 MIG, 然后通过一些规则对 MIG 进行优化的逻辑综合方法. 文献[10]提出利用 M 门的容错性, 通过引入适当的逻辑错误实现逻辑优化, 但由于待引入逻辑错误的选择需要考虑对其他输入的影响, 因此优化过程比较复杂. 文献[11]提出将用 MIG 形式表示的逻辑函数来进行 FPGA 综合, 利用 FPGA 实现 MIG 的优化技术. 文献[9-11]中得到的

MIG 都是从 AOIG 转换过来的, 相比于逻辑函数的 AOIG 形式, AIG 形式应用更加普遍, 因此本文将

以 AIG 为基础实现基于 M 门映射的电路面积优化. 本文优化方法将结合等效替代和电路覆盖策略, 用 M 门和反相器实现的电路面积优化. 主要包括两方面内容: (1)利用 M 门和传统的“与”“或”门之间的等价关系, 以及 M 门的 AIG 实现形式, 实现 M 门对传统布尔逻辑中“与”“或”运算的替代和对 AIG 子图的替换. (2)利用基于动态规划的电路覆盖算法, 完成对 AIG 切割, 并使得切割后子 AIG 的集合有利于 M 门的映射, 最终实现以面积优化为导向的用 M 门映射的逻辑电路.

1 逻辑函数的 Majority 运算实现

1.1 M 门的运算性质

M 门的逻辑功能可用下式来表示,

$$\langle a, b, c \rangle = ab + bc + ac. \quad (1)$$

其中, 变量 a, b 和 c 为 3 个输入.

(1)式中 $\langle a, b, c \rangle$ 表示输入为 a, b, c 的 M 门的逻辑功能. 由(1)式可知, 当 3 个输入中, 有 2 个及以上输入为逻辑“1”时, 输出为逻辑“1”; 反之, 当有 2 个及以上输入为逻辑“0”时, 输出为逻辑

收稿日期: 2018-01-16.

宁波大学学报(理工版) 网址: <http://journalg.nbu.edu.cn/>

基金项目: 国家自然科学基金(61471211).

第一作者: 吴达斌(1993-), 男, 安徽宿州人, 在读硕士研究生, 主要研究方向: 逻辑综合与优化. E-mail: 1611082561@nbu.edu.cn

*通信作者: 王伦耀(1972-), 男, 浙江宁波人, 教授, 主要研究方向: 逻辑综合与优化. E-mail: wanglunyao@nbu.edu.cn

“0”即M门逻辑功能相当于一个多数表决器。此外, 当M门中有1个输入为0时, 如 $a=0$ 时, (1)式就变成下式:

$$\langle 0, b, c \rangle = 0 \cdot b + bc + 0 \cdot c = bc. \quad (2)$$

(2)式中的M门逻辑功能相当于二输入的“与”门; 同理, 当M门中有一个输入为1时, 如 $a=1$ 时, (1)式就变成(3)式, 此时M门的逻辑功能相当于二输入的“或”门。

$$\langle 1, b, c \rangle = 1 \cdot b + bc + 1 \cdot c = b + c. \quad (3)$$

由(2)式和(3)式可知, 将M门和反相器组合在一起就可以构成一个逻辑完备集, 可用来表示任何的逻辑函数。

对于一个有 n 个输入变量, m 个乘积项构成的逻辑函数 $f(X)$, $X = \{x_1, x_2, \dots, x_n\}$, 其对应的乘积项之和(SOP)形式为 $f(X) = \sum_{i=1}^m P_i$, 利用式(2)可以将乘积项 $P_i(X)$ 表示成:

$$P_i(X) = M_1(x_1, x_2, 0) \cdot M_2(x_3, x_4, 0) \cdots \cdots, \\ M_{n/2}(x_{n-1}, x_n, 0), \quad (4)$$

或

$$P_i(X) = M_1(x_1, x_2, 0) \cdot M_2(x_3, x_4, 0) \cdots \cdots \\ M_{(n-1)/2}(x_{n-2}, x_{n-1}, 0) \cdot x_n. \quad (5)$$

(4)式中的 n 为偶数, (5)式中的 n 为奇数。循环利用(2)式, 最终可以把(4)式和(5)式中的“与”运算替换成M运算。同理, 循环利用(3)式, 可以把 $f(X) = \sum_{k=1}^m P_k$ 中的“或”运算替换成M运算。最终 $f(X) = \sum_{k=1}^m P_k$ 可以转换为M运算和取反运算构成的表达式。

1.2 逻辑函数的MIG表示

MIG是由M门和反相器组成的一种数据结构。逻辑函数可通过相应的运算定律进行转换和化简, MIG和逻辑函数类似, 也有相应运算法则。MIG的布尔代数运算法则有2个, 分别是图1的 Ω 法则^[9]

交换律: $\langle x, y, z \rangle = \langle y, x, z \rangle = \langle z, y, x \rangle$

多数律: $\begin{cases} \text{if } (x = y) : \langle x, x, z \rangle = \langle y, y, z \rangle = x = y \\ \text{if } (x = \bar{y}) : \langle x, \bar{x}, z \rangle = z \end{cases}$

结合律: $\langle x, u, \langle y, u, z \rangle \rangle = \langle z, u, \langle y, u, x \rangle \rangle$

分配律: $\langle x, y, \langle u, v, z \rangle \rangle =$

$$\langle \langle x, y, u \rangle, \langle x, y, v \rangle, z \rangle$$

取反律: $\overline{\langle x, y, z \rangle} = \langle \bar{x}, \bar{y}, \bar{z} \rangle$

图1 Ω 法则

和图2的 Ψ 法则^[10]。

相交律: $\langle x, y, z \rangle = \langle x, y, z_{x/y} \rangle$

互补律: $\langle x, u, \langle y, \bar{u}, z \rangle \rangle = \langle x, u, \langle y, x, z \rangle \rangle$

替代律: $\langle x, y, z \rangle = \langle v, \langle \bar{v}, \langle x, y, z \rangle_{v/\bar{u}} \rangle, u \rangle,$
 $\langle \bar{v}, \langle x, y, z \rangle_{v/\bar{u}}, \bar{u} \rangle$

图2 Ψ 法则

以图3为例, 逻辑函数 $f = A(B + \bar{B}C)$ 可通过(2)式和(3)式得到 $\bar{B}C = \langle \bar{B}, 0, C \rangle$, $B + \bar{B}C = \langle B, 1, \langle \bar{B}, 0, C \rangle \rangle$, $A(B + \bar{B}C) = \langle A, 0, \langle B, 1, \langle \bar{B}, 0, C \rangle \rangle \rangle$, 对应的MIG结构如图3(a)所示。图3(a)通过 Ψ 法则的互补律, 即把第3层的输入0用第2层的输入 B 替换, 得到图3(b), 再通过 Ω 法则的多数律, 即第3层的2个输入 B 和 $\bar{B}$ 互为反变量, 输出可以由第3个输入 C 表示, 得到图3(c), 即最终简化的MIG。逻辑函数 $f = A(B + \bar{B}C)$ 通过传统公式法可化简为 $f = AB + AC$, 与MIG的布尔代数运算法则得到的结果一致, 即化简得到的结果正确。

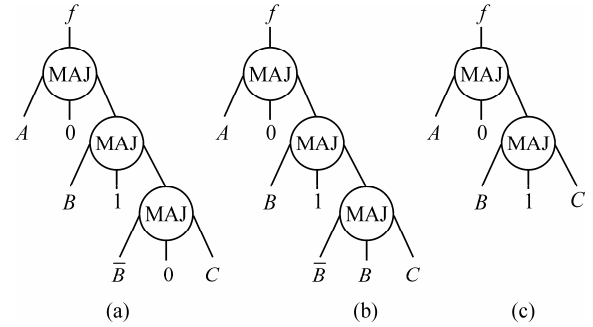


图3 逻辑函数的转化和优化

逻辑电路的逻辑功能除了用逻辑函数来描述外, 也可用AIG数据结构来表示。AIG中只有“与”和“非”2种逻辑运算, 图4中每个圈表示1个“与”运算节点, 实边表示输入或输出变量, 其中虚边表示对应的变量取反。图4(a)是 $f = ABC + \bar{A}\bar{B}C$ 对应

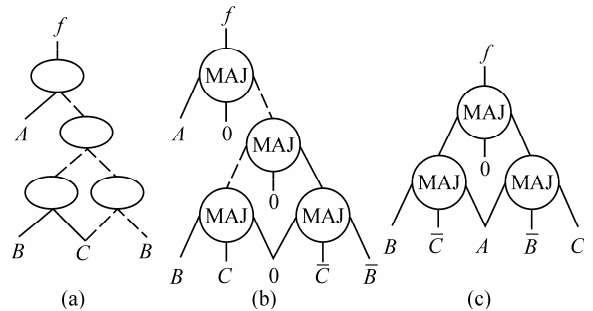
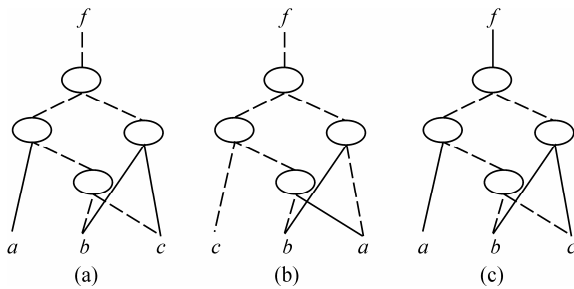


图4 AIG到MIG的转化和优化

的 AIG 表示, 根据(2)式可得, 与门可用 M 门实现, 用 M 门替代图 4(a)中与门就得到图 4(b)中的 MIG 图, 对图 4(b)使用按照 Ω 法则的取反律、多数律以及 Ψ 法则的互补律来优化, 得到图 4(c)中的 MIG. 图 4(c)为 $y = \langle \langle A, B, \bar{C} \rangle, 0, \langle A, \bar{B}, C \rangle \rangle$, 对应的逻辑表达式为 $y = (AB + \bar{A}\bar{C} + \bar{B}\bar{C})(\bar{A}\bar{B} + AC + \bar{B}C)$, 对此逻辑表达式化简可得 $f = ABC + \bar{A}\bar{B}\bar{C}$. 由此可知, MIG 的优化结果正确.

2 基于 Majority 门映射的面积优化

考虑到逻辑综合中, AIG 是一种普遍使用的表示逻辑功能的数据结构, 因此本文基于 M 门映射的面积优化将以 AIG 为基础. 在 AIG 向 MIG 转化时, 可以采用(2)式所示方式, 将 AIG 中的与门用 M 门代替, 也可以利用 M 门逻辑功能与 AIG 结构对应的逻辑功能的等效性, 直接将 AIG 的子结构替换成 M 门. 对于三输入的 M 门, 在输入变量存在反变量情况下, 可产生 8 种变换形式; 就输入变量次序而言, 可产生 6 种变换形式; 另外输出也存在取反和不取反 2 种情况, 所以对一个三输入的 M 门一共可以产生 96 种能够让 AIG 结构节点数发生变化的变换形式. 图 5 即给出了几种在不同输入情况下 M 门对应的 AIG 等价逻辑结构.



(a) $\langle a, b, c \rangle$ 对应的 AIG 结构, (b) $\langle \bar{c}, b, \bar{a} \rangle$ 对应的 AIG 结构, (c) $\langle \bar{a}, b, \bar{c} \rangle$ 对应的 AIG 结构.

图 5 M 门用 AIG 实现的等价逻辑结构

本文基于 M 门映射的面积优化大体包含下面 3 个部分: (1)通过对 AIG 的切割, 得到子 AIG 的集合; (2)将子 AIG 对应的逻辑功能用 M 门和反相器的组合实现, 即将子 AIG 转化为子 MIG; (3)使用 Ω 法则和 Ψ 法则对转换后的 MIG 进行以面积为导向的优化. 由于不同切割得到的子 AIG 对生成的 MIG 面积有很大影响, 因此, 本文提出的面积优化方法是通过搜索不同切割形成不同子 AIG 的集合

以及对应的 MIG, 最终得到有利于电路面积优化的 MIG, 进而实现基于 M 门映射的面积优化, 其中最优化搜索采用动态规划算法. 基于 M 门映射的电路面积优化算法的伪代码如图 6 所示.

```

Algorithm MIG_mapping(AIG_nodes)
input AIG_nodes;
if (Leaf_Node(Node n)) {
    return 0; }
initial n.bestarea;
Cut AIG;
do { //loop1
     $C_j \rightarrow \text{MIG}$ ;
    do { //loop2
         $\Omega(\text{MIG})$ ;
         $\Psi(\text{MIG})$ ; }
    while (no improvement)
     $S_j = \text{Majority numbers}$ ;
    if ( $n.\text{bestarea} > S_j$ ) {
         $n.\text{bestarea} = S_j$ ; }
     $j++$ ; }
while(convert all sub-AIGs to sub-MIGs)
output n.bestarea.

```

图 6 最优面积覆盖的优化算法

在上述算法中, 首先需判断所选取节点是否为叶子节点, 若是, 则不需要对其进行切割; Cut AIG 表示对 AIG 进行切割, 对 AIG 切割时, 必须满足 FFC(Fanout Free Cone)的约束条件, 即以节点 n 为根节点的最大 FFC 进行切割^[12-13]. 在切割过程中, 若得到的子 AIG 出现了不满足 FFC 的约束条件, 则将此切割视为无效切割, 不使用 M 门对其进行覆盖. 此外, 若 AIG 中出现某个节点输出的同时被作为多个节点的输入时, 则考虑这种节点有助于电路的面积优化, 文中将此类节点表示为 AIG 的公共节点. 算法会将公共节点作为一个子 AIG 的根节点来处理. 此外, 算法中存在 2 个主要的循环, 第 1 个循环是用来表示遍历 AIG 的所有切割, 当每种切割都转换后结束循环; 第 2 个循环用来表示每种切割下得到的 MIG 使用 Ω 法则和 Ψ 法则进行优化, 直到 MIG 中用 Ω 法则和 Ψ 法则对 MIG 不能进行优化后结束循环.

算法中 $C_j \rightarrow \text{MIG}$ 来表示 AIG 向 MIG 转换. 本文中根据每种切割得到的子 AIG 输入变量个数

不同进行如下转换:

(1)子 AIG 输入变量的个数小于 3 时, 不满足本文提出的能够减少电路节点个数的子 AIG 结构, 可直接用(2)式将子 AIG 结构的与门用 M 门替换.

(2)子 AIG 输入变量的个数等于 3 时, 搜索对应子 AIG 的内部结构, 看它是否与一个三输入的 M 门对应 AIG 的内部结构一致. 若 AIG 的内部结构为上述的 96 种情形中的一种, 就用一个三输入的 M 门直接替换子 AIG, 否则就用(2)式将子 AIG 结构的与门用 M 门替换.

(3)子 AIG 当输入变量的个数大于 3 时, 对其输入变量个数等于 3 的 AIG 进行任意搜索, 若满足 M 门对应 AIG 的内部结构, 就用一个 M 门直接替换, 其他的与门用 M 门进行一对一的替换, 若找不到满足 M 门对应 AIG 的内部结构, 则将子 AIG 结构的与门用 M 门替换.

算法中 $\Omega(\text{MIG})$ 和 $\Psi(\text{MIG})$ 是利用图 1 和图 2 所示的 Ω 法则和 Ψ 法则, 对得到的初始 MIG 进行化简.

图 6 算法的功能是求取一个电路的最优面积覆盖. 输出值 $n.\text{bestarea}$ 是关于节点 n 的最大 FFC 的最优覆盖面积, 且面积用 M 门的个数来衡量. 而所有最大 FFC 最优覆盖的面积相加, 就是整个电路的最优覆盖下的面积.

以图 7 所示电路为例, 说明本文算法优化过程. 图 7 为同个电路的 AIG 通过不同切割得到的覆盖, 其中每个实线小椭圆代表 1 个与门, 连接与门的虚线边表示取反运算. 电路由 6 个与门组成, 分别记为 n, p, q, r, s, t , 电路有 4 个输入, 分别为 a, b, c, d . 图 7 中的每个虚线框代表 1 个切割, 如图 7(a)中一种得到了 3 个切割, 分别用 C_1, C_2, C_3 表示.

(1)以图 7(a)为例, 执行 Cut AIG 后, 得到图 7(a)中分别以节点 n, s, t 为根节点得到的 3 个切割 C_1, C_2 和 C_3 . 首先判断切割 C_1, C_2, C_3 都满足约束 FFC, 为有效切割; 其中 C_2, C_3 的 s 和 t 节点的输出被 r 和 q 节点引用, 都为公共节点, 有利于面积优化, 应保留. 因此把 s 和 t 作为子 AIG 的根节点来处理. 如果执行 Cut AIG 后, 得到图 7(c)所示的切割, 考虑到切割 C_3 的输出来自 2 个内部节点, 不满足约束 FFC, 因此图 7(c)所示覆盖为无效覆盖, 应放弃.

(2)将有效切割得到每个子 AIG 转换到子 MIG.

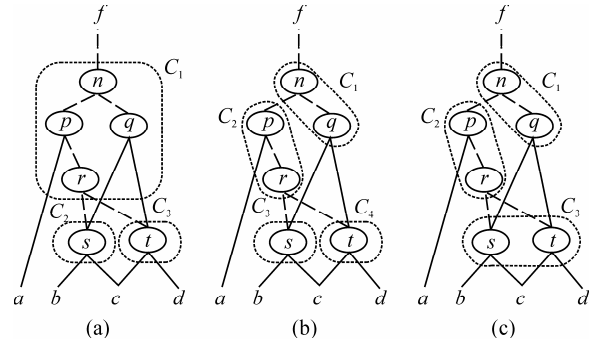


图 7 同一电路的不同覆盖

根据子 AIG 转换到 MIG 的操作步骤, 对于图 7(a)中子 AIG 切割 C_1 , 其输入变量个数为 3 个, 则搜索切割 C_1 的内部结构, 发现其 M 门对应的 AIG 结构一致, 用一个 M 门替换切割 C_1 ; 对于切割 C_2 和 C_3 , 判断出其输入变量的个数小于 3 个, 则直接将 2 个切割中的与门用 M 门进行替换. 将每个子 AIG 全部转换到 MIG 后, 就得到了初始 MIG, 过程如图 8(a)所示.

(3)利用 Ω 法则和 Ψ 法则对转换后的 MIG 进行简化. 对图 8(a)所示的初始 MIG 进行简化后, 最终得到如图 8(b)所示的 MIG. 计算简化后 MIG 中 M 门的个数, 得到该有效切割下电路的最优覆盖面积为 2.

(4)图 7(b)为另外一种满足约束条件的一个有效覆盖, 重复(1)~(3), 得到不同覆盖对应的 M 门的个数. 比较不同覆盖对应的电路面积, 最终得到最优的覆盖面积为 2.

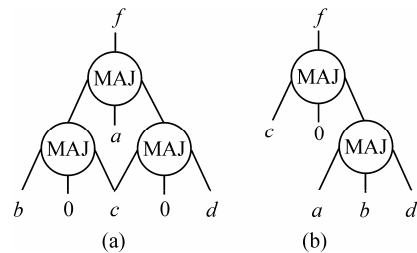


图 8 切割后的覆盖和优化过程

3 实验结果及分析

本文提出的基于 Majority 逻辑门映射的电路面积优化算法用 C 语言编程, 并在 WIN7 64 位操作系统 4 GB 内存中实现. 考虑到 AIG 或 MIG 内部的节点对应与门或者 M 门, 并且在量子计算中实现 M 门的成本和实现 AND/OR 的成本一致^[14], 因

此可以使用节点个数来衡量电路面积大小^[15]. 本文将 MCNC 电路作为测试电路, 覆盖搜索过程中限定生成切割得到的子 AIG 最大输入个数为 5. 表 1 为本文方法与文献[16]中逻辑综合工具 ABC 比较的结果. 表 1 的第 1 列和第 2 列为测试电路的名称和电路的输入和输出以及 PLA 表示的乘积项个数; 表 1 中 AIG 节点数和 MIG 节点数分别用 ABC 工具和本文算法后得到的节点数. 表 1 的最后一列表示用本研究方法和 ABC 比较的结果, 其计算方法如(6)式.

表 1 MCNC 电路测试结果

Benchmark		Node	Node	Imp/%
circuit	i/o/p	(AIG)	(MIG)	
adder	256/128/1 020	1 019	892	-12.46
5xp1	7/10/131	97	88	-9.28
9sym	9/1/238	213	181	-15.02
apex1	45/45/2 604	1 975	1 698	-3.90
apla	10/12/273	182	176	-3.30
b2	16/17/1 629	1 186	1 025	-13.57
bar	135/128/3 336	3 141	2 725	-13.24
bca	26/46/3 103	2 250	2 063	-8.31
ex5	8/63/1 193	501	440	-12.18
i10	257/224/2 675	1 829	1 716	-6.18
in1	16/17/1 629	1 563	1 025	-34.42
max	512/130/2 865	2 834	2 362	-16.65
chkn	29/7/431	344	339	-1.45
sin	24/25/5 416	5 039	4 719	-6.35
Average		1 583	1 389	-12.29

$$\text{Imp} = \frac{\text{Node(MIG)} - \text{Node(AIG)}}{\text{Node(AIG)}} \times 100\%. \quad (6)$$

从表 1 实验结果数据可以看出, 与逻辑综合工具 ABC 得到的 AIG 电路相比, 本文提出的基于 M 门映射的算法后得到的电路具有更小面积, 节点数平均减少了 12.29%. 但不同电路用 M 门来实现, 其电路面积优化的效果也不同. 表 1 中电路 chkn、apex1、apla 等电路用 M 门来实现后, 电路优化结果不明显, 节点数最少的只改进了 1.45%, 而 in1 电路优化效果明显, 达到 34.42%. 上述优化结果的差异与电路对应的 AIG 结构密切相关. 如果电路对应的 AIG 中找不到适合 M 门表示的结构, 或者 AIG 转化为 MIG 后, 很难用 Ω 法则和 Ψ 法则进

行进一步化简, 对于这种 AIG 几乎就是用(2)式将 AIG 中的与节点转换为 MIG 中的 M 节点, 导致简化后的 MIG 节点数与 AIG 节点数差别不大. 反之, 如电路对应的 AIG 中, 存在比较多适合 M 门实现的结构, 则优化效果就比较好. 以 chkn 为例, 其对应的 AIG 中, 找到如图 5 所示的可以直接替换成 M 门的结构一共有 3 处, 而电路 in1 一共找到了直接替换成 M 门的结构一共有 97 处. 此外, 由替换后剩余的 AIG 得到的 MIG, 电路 in1 对应的 MIG 也更适合用 Ω 法则和 Ψ 法则进行进一步化简. 总的来说, 由于 MIG 使用 M 门比 AIG 使用与门的逻辑功能更加丰富, 能够表示一些复杂的逻辑函数, 利用 MIG 形式表示电路在面积优化方面相比于利用 AIG 形式表示电路更加具有优势.

4 结论

本文提出基于三输入的 M 门映射的电路面积优化方法. 该优化方法利用适合 M 门实现 AIG 子结构的搜索, M 门与传统与门、或门逻辑功能的等效变换, AIG 的切割、适合 M 门映射的覆盖搜索等将 AIG 转化为适合 M 门映射的 MIG, 然后利用 Ω 法则和 Ψ 法则对 MIG 进行进一步简化, 从而实现基于 M 门映射的电路面积优化. 实验结果表明, 相比于逻辑综合工具 ABC 结果, 用 M 门映射后可得到平均约 12%左右的电路面积优化; 但就单个电路而言, 电路面积优化程度与其 AIG 结构密切相关, 其优化程度介于 1.45%~34.42%之间.

参考文献:

- [1] De Marchi M, Sacchetto D, Frache S, et al. Polarity control in double-gate, gate-all-around vertically stacked silicon nanowire FETs[EB/OL]. [2017-10-15]. <https://si2.epfl.ch/~demichel/publications/archive/2012/S08P04.pdf>.
- [2] Marchi M D, Zhang J, Frache S, et al. Configurable logic gates using polarity-controlled silicon nanowire gate-all-around FETs[J]. IEEE Electron Device Letters, 2014, 35(8):880-882.
- [3] Appenzeller J. Carbon nanotubes for high-performance electronics: Progress and prospect[J]. Proceedings of the IEEE, 2008, 96(2):201-211.
- [4] Yang H, Heo J, Park S, et al. Graphene barristor, a triode device with a gate-controlled Schottky barrier[J]. Science, 2012, 336(6085):1140-1143.

- [5] Li S L, Miyazaki H, Lee M V, et al. Complementary-like graphene logic gates controlled by electrostatic doping[J]. Small, 2011, 7(11):1552-1556.
- [6] Fackenthal R, Kitagawa M, Otsuka W, et al. 19.7 A 16Gb ReRAM with 200 MB/s write and 1 GB/s read in 27 nm technology[C]. IEEE, Solid-State Circuits Conference Digest of Technical Papers, 2014:338-339.
- [7] Amlani I, Orlov A O, Toth G, et al. Digital logic gate using quantum-dot cellular automata[J]. Science, 1999, 284(5412):289-291.
- [8] Cornil J, Brédas J, Zaumseil J, et al. Ambipolar transport in organic conjugated materials[J]. Advanced Materials, 2010, 19(14):1791-1799.
- [9] Gaillardon P E, Micheli G D. Majority-inverter graph: A novel data-structure and algorithms for efficient logic optimization[C]. ACM Design Automation Conference, 2014:1-6.
- [10] Amaru L, Gaillardon P E, Micheli G D. Majority-inverter graph: A new paradigm for logic optimization[J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2016, 35(5):806-819.
- [11] Amaru L, Petkovska A, Gaillardon P E, et al. Majority-inverter graph for FPGA synthesis[C]. The Workshop on Synthesis & System Integration of Mixed Information Technologies, 2015.
- [12] 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(理工版), 2016, 29(4):38-43.
- [13] 岑旭梦, 王伦耀, 夏银水, 等. Library-free 映射在电路面积优化中的应用[J]. 计算机辅助设计与图形学学报, 2017, 29(11):2147-2152.
- [14] Soeken M, Roetteler M, Wiebe N, et al. Design automation and design space exploration for quantum computers[EB/OL]. [2017-05-10]. <http://www.microsoft.com/en-us/research/wp-content/uploads/2017/05/1612.00631.pdf>.
- [15] Chen Y C, Wang C Y. Logic restructuring using node addition and removal[J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2012, 31(2):260-270.
- [16] Berkeley Logic Synthesis and Verification Group. ABC: A system for sequential synthesis and verification[EB/OL]. [2017-04-12]. <https://bitbucket.org/alanmi/abc>.

Area optimization using Majority logic gate mapping

WU Da-bin, WANG Lun-yao*, XIA Yin-shui

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

Abstract: An area optimization algorithm for logic circuits using 3-inputs Majority logic gate (M gate) and inverter mapping is proposed in this paper. By cutting the AIG with a data structure for logic circuit into sub-AIGs, searching the patterns in sub-AIGs which is suitable for M gate mapping and replacing the “AND” nodes with M nodes, the algorithm converts an AIG to an MIG (Majority-Inverter Graph). Furthermore, by employing the Ω and Ψ rules, a more compacted MIG with fewer nodes is obtained, which facilitates further area reduction. The proposed algorithm is implemented in C programming language and tested under MCNC benchmarks. The experimental results show that compared with the result of ABC, the average area reduction of the tested circuits is about 12% using the M gates mapping.

Key words: Majority logic gate; mapping algorithm; logic optimization

(责任编辑 章践立)