

基于逻辑复合门映射的电路面积优化

岑旭梦, 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: 提出了一种基于复合门映射的电路面积优化方法. 该方法首先通过 BDD 结构的拆分将待优化的逻辑电路转化为由二输入基本逻辑门组成的多级逻辑网络, 然后利用复合门搜索和等效转换算法将逻辑网络中适合复合门实现的子结构用相应的复合门替换. 提出的优化算法用 C 语言编程实现, 并用 MCNC 电路进行了测试. 测试结果表明, 相比于 ABC 和 BDS 软件结果, 经算法优化后, 电路的节点数可分别减少约 37% 和 29%.

关键词: 复合门; 映射; BDD; 逻辑优化

中图分类号: TN47; TP391

文献标志码: A

文章编号: 1001-5132 (2016) 04-0038-06

逻辑映射是逻辑综合中一个重要环节, 并直接影响电路的面积、功耗、速度等方面的优化. 逻辑映射可以通过调用单元库(Cell Library)中单元电路方式实现, 也可以采用自由库(Library Free)方式来实现^[1-4]. 基于单元库的映射, 其优化效果与库中包含的逻辑单元电路种类以及单元电路性能直接相关. 显然, 当单元库中包含的逻辑单元越丰富, 单元电路的性能越好, 映射的效果就越好. 但单元库中包含的单元电路数量越多, 搜索时间就越长, 从而影响映射效率. 而基于自由库的映射可根据电路实际, 动态地生成适合的单元电路, 有助于映射后的电路面积优化和性能的提升^[5], 因此越来越受到研究者的关注. 在基于自由库映射的研究中, 文献[2]通过搜索电路网络中满足单元约束条件的所有割集, 并将割集转化为可能的逻辑单元电路, 来完成逻辑电路基于自由库的映射. 文献[3]提出了一种基于有向无环图(Directed Acyclic Graph, DAG)的映射方法, 该方法先将电路表示为 DAG, 然后利用 DAG 重组方法产生相应的逻辑单元电路, 进而实现电路的映射. 文献[4]提出了基于 DAG 分解的映射方法, 在映射过程中通过将分解得到的 DAG 子图用相应的逻辑单元来替换, 实现电路的映射.

本文主要研究基于自由库的映射中逻辑复合门在电路面积优化中的应用. 考虑到在单元库的映射里基本单元电路一般以反相器、二输入和多输入的与非门、或非门等基本逻辑门为主, 而很少涉及复合逻辑门. 但有研究表明^[6-7], 相比于二输入基本逻辑门, 三输入逻辑复合门包含更多复杂的逻辑运算, 同时在实现相同逻辑功能情况下, 逻辑复合门具有更简单的电路结构. 逻辑复合门的这种特性为实现电路面积的优化提供了可能. 本研究中逻辑复合门映射在电路面积优化中的应用主要内容为: (1)将待映射的逻辑电路用二元决策图(Binary Decision Diagram, BDD)^[8-11]表示, 通过对 BDD 拆分和分析, 完成适合逻辑复合门实现的结构搜索与构造, 并用电路节点个数作为判别电路大小的依据. (2)采用不同逻辑复合门之间的等效变换方法, 将原电路中不同种类的复合门用同一种复合门实现, 并保持逻辑功能不变.

1 适合复合门实现的逻辑结构搜索与构造

在进行复合门逻辑映射前, 首先要确定逻辑网络中哪些结构适合用复合门实现, 即适合复合门实现的逻辑结构搜索; 然后将该逻辑网络用逻辑复合门等效替代, 从而实现电路节点的减少. 复

收稿日期: 2015-11-16.

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(61131001, 61471211).

第一作者: 岑旭梦(1992-), 男, 浙江宁波人, 在读硕士研究生, 主要研究方向: 数字集成电路逻辑综合和优化. E-mail: cenxumeng@qq.com

*通信作者: 王伦耀(1972-), 男, 浙江宁波人, 教授, 主要研究方向: 数字集成电路逻辑综合和优化. E-mail: wanglunyao@nbu.edu.cn

合门虽然可以有效地简化电路形式, 但是随着输入的增加, 复合门内部电路将变得复杂, 其负载电容也会大大增加, 从而影响复合门的性能. 因此, 复合门的输入往往限于 3 至 4 个, 本研究主要对适合三输入复合门实现的电路结构进行搜索, 并用相应的复合门实现. 复合门搜索是在由“与”、“或”、“同或”等基本逻辑单元构成的多级逻辑网络中进行. 该多级逻辑网络可利用 Yang 等^[12]提出的 BDD 的拆分方法实现.

在由基本逻辑门构成的多级逻辑网络中, 相连的逻辑门之间的不同组合可构成不同的逻辑复合门, 如图 1 所示. 在图 1 中每个圈中的二个基本逻辑门都可以构成一个逻辑复合门. 但是在复合门构造过程中, 存在不同复合门同时调用同一个逻辑门的现象, 如复合门 4、复合门 5 和复合门 6 同时调用了门 M6. 而在实际逻辑综合中, 一旦门 M6 与逻辑门 M3 构成了复合门 4 后, M6 就不能再同门 M4 构成复合门 5 或者同门 M7 构成复合门 6. 另一方面, 当某个逻辑网络子结构可以由多种复合门实现时, 还需要考虑不同复合门的性能, 以使电路性能得到提升. 因此, 在用复合门进行映射时, 先要对待优化逻辑网络进行搜索, 以此用来确定待优化的逻辑网络中哪些结构适合用哪些复合门实现.

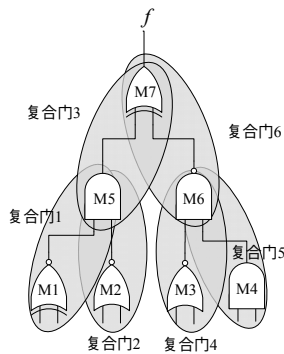


图 1 适合复合门实现的逻辑结构搜索

Algorithm 1 是本文提出的搜索算法的伪代码, 其中变量 HABLE 是为防止重复搜索而建立的哈希表; RECOBLE 用来记录各种结构数量; 伪代码中 Generator()用于将实际结构转化为 RECOBLE 中对应的入口地址. 通过对 RECOBLE 表记录的各种结构数量分析, 可以得到比例最高的复合结构, 该复合结构即为表 1 所示的搜索结果.

Algorithm 1 complex struct detection

Input: HABLE, node, RECOBLE

Function: StructDetect

if node $\in$ HABLE $\parallel$ node is terminal node **then**
return

else

HABLE $\leftarrow$ node

end if

value0 $\leftarrow$ Generator(node, fanin0(node))

RECOBLE $\leftarrow$ value0

value1 $\leftarrow$ Generator(node, fanin1(node))

RECOBLE $\leftarrow$ value1

StructDetect(HABLE, fanin0(node), RECOBLE)

StructDetect(HABLE, fanin1(node), RECOBLE)

以图 1 为例, 若 M7 为根节点, 首先判断 M7 是否是原始输入, 若不是, 则将 M7 与 M5 组合成为复合门 3, 并加入到表 RECOBLE 中; 否则, 停止该搜索. 接着, 将 M7 与 M6 组合, 得到复合门 6, 并加入到 RECOBLE 中. 然后, 分别以 M5 和 M6 为根节点重复以上过程. 当对电路中所有节点都访问一次后, 搜索结束. 最终在 RECOBLE 中可以得到所有可能的复合门, 对应复合门 1~6. 统计 RECOBLE 结果, 可以发现图 1 中 AND-NOR 结构最多. 表 1 给出了搜索算法对部分 MCNC 电路进行复合门的搜索结果.

表 1 部分复合门结构探测结果

电路名	最多的结构	OR-AND/%	OR-NOR/%	XNOR-OR/%	AND-OR/%
5xp1	OR-AND	16	9	1	11
9sym	OR-AND	13	11	0	10
alu4	OR-NOR	14	15	1	9
apex3	OR-NOR	12	15	0	8
b12	OR-NOR	13	16	1	5
clip	OR-NOR	10	12	0	9
con1	OR-AND	13	13	0	8
cordic	OR-AND	9	4	5	8
t481	XNOR-OR	0	0	28	0
平均		11.1	11.6	3.9	7.6

表 1 中复合门的百分比表示整个电路中指定复合门个数与总的复合门个数之比. “最多的结构”这一列表示在一个电路中出现最频繁的复合门.

从表1可以发现, 电路中OR-AND结构和OR-NOR结构是大多数电路中最多的结构, 平均为11.1%和11.6%. 利用Algorithm 1确定复合门种类后, 就可以用Algorithm 2实现用复合门来替代逻辑网络中相应子结构.

Algorithm 2 combination procedure

Input: *node*, *SHAREDTABLE*, *destiststructure*

Function: StructCombine

if *node* is terminal node **then**

 return

end if

child0 ← Fanin0(*node*)

child1 ← Fanin1(*node*)

StructCombine(*child0*, *SHAREDTABLE*,
destiststructure)

StructCombine(*child1*, *SHAREDTABLE*,
destiststructure)

if (*node*, Fanin0(*node*)) match *destiststructure* **then**

 transfer (*node*, Fanin0(*node*)) to one
 complex node

end if

if (*node*, Fanin1(*node*)) match *destiststructure* **then**

 transfer (*node*, Fanin1(*node*)) to one
 complex node

end if

Algorithm 2 伪代码中 SHAREDTABLE 表示被共用节点的汇总表; destiststructure 表示目标复合门. 以图2为例, 假设目标复合门为 AND-NOR, 从最底端开始, 判断当前节点与其中扇入相连的节点是否与目标结构匹配, 若匹配则将这2个节点映射为一个复合门. 在图2(a)中 M5 和 M1 与目标复合门不匹配, 所以检查 M5 与另一个扇入, M5 与 M2 构成 AND-NOR, 得到复合门2. 同样对 M6 的2个扇入判断, 可得到复合门4与复合门5'. 其中复合门5' 不是由2个二输入门直接合并得到, 而是将 M4 逻辑等效为 M4' 后与 M6 组合而成. 考虑到 M6 被复合门4与复合门5' 重复调用, 所以复合门4与复合门5' 之间只能保留其中一个. 在本算法中, 搜索是按从左至右顺序进行的, 因此最终得到的应该是复合门4. 同理, 再对上级电路进行复合门的构造, 最终可得到图2(b)结构的包含三输入复合门

的电路, 虚线引出的部分为另一种可行电路.

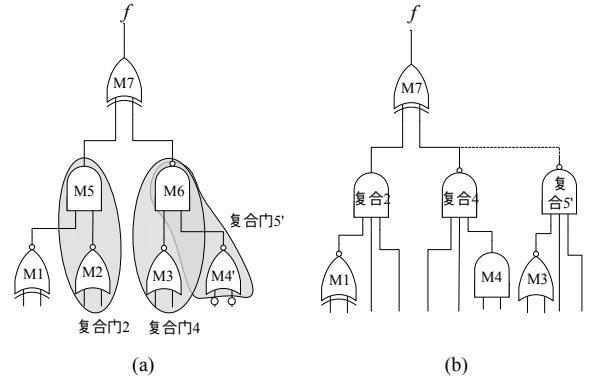


图2 逻辑结构的复合门替代

2 复合门之间的等效变换

在适合复合门实现的结构搜索过程中, 可以通过逻辑“与”和逻辑“或”以及逻辑“异或”和逻辑“同或”之间的等效变换, 实现不同逻辑复合门之间的等效变换. 图1中的M4和图2(a)中的M4'就是等效变换的结果. 本研究用到的逻辑门之间等效变换的规则如下:

规则1 二输入“与”门和二输入“或”门之间的相互转化时, 转化前后2个门输入与输出端口逻辑取反的个数之和为3, 且2种类型门的取反端口互不重合.

证明 (1)在“与”门转换成“或”门时规则1成立. 设 f 是“与”门的输出, 变量 $\dot{a}$ 和 $\dot{b}$ 是“与”门的输入, 其中 $\dot{a}$ 表示 a 或 $\bar{a}$; $\dot{b}$ 表示 b 或 $\bar{b}$. 由摩根定律得:

$$f = \dot{a}\dot{b} = \overline{\overline{\dot{a}}\overline{\dot{b}}} = \overline{\overline{\dot{a}} + \overline{\dot{b}}} \quad (1)$$

由式(1)可得, 无论 $\dot{a}$ 和 $\dot{b}$ 取原变量还是反变量, 在逻辑“与”表达式 $\dot{a}\dot{b}$ 和逻辑“或”表达式 $\overline{\overline{\dot{a}} + \overline{\dot{b}}}$ 中, 取反符号“ $\bar{}$ ”的个数总和恒等于3; 并且当表达式 $\dot{a}\dot{b}$ 中某个变量或输出取反变量时, 则在 $\overline{\overline{\dot{a}} + \overline{\dot{b}}}$ 中该变量或输出一定取原变量, 即2种类型门的取反端口互不重合.

(2)在“或”门转换成“与”门时规则1也成立. 设 g 是“或”门的输出, 变量 $\dot{c}$ 和 $\dot{d}$ 是“或”门的输入, 其中 $\dot{c}$ 表示 c 或 $\bar{c}$; $\dot{d}$ 表示 d 或 $\bar{d}$. 由摩根定律得:

$$g = \dot{c} + \dot{d} = \overline{\overline{\dot{c}}\overline{\dot{d}}} = \overline{\overline{\dot{c}}}\overline{\overline{\dot{d}}} = \dot{c}\dot{d} \quad (2)$$

由式(2)可得到, 无论 $\dot{c}$ 和 $\dot{d}$ 取原变量还是反变量, 在逻辑“或”表达式 $(\dot{c} + \dot{d})$ 和逻辑“与”表达式 $\overline{\dot{c}\dot{d}}$ 中, 取反符号“-”的个数总和恒等于 3; 并且当表达式 $(\dot{c} + \dot{d})$ 中某个变量或输出取反变量时, 则在 $\overline{\dot{c}\dot{d}}$ 中该变量或输出一定取原变量. 即 2 种类型门的取反端口互不重合.

综上可得, 规则 1 成立. 证毕.

规则 2 对二输入“异或”门或者二输入“同或”门输入输出端口取反时, 如 3 个端口取反的个数为偶数个时, 可以同时将这些取反操作忽略; 当 3 个端口取反的个数为奇数个时, 等效为对任一端口的取反.

证明 设 f 是“异或”门的输出, 变量 $\dot{a}$ 和 $\dot{b}$ 是“异或”门的输入, 其中 $\dot{a}$ 表示 a 或 $\bar{a}$; $\dot{b}$ 表示 b 或 $\bar{b}$. 变量 $\dot{a}$ 和 $\dot{b}$ 之间的“异或”满足式(3):

$$f = \dot{a} \oplus \dot{b} = \dot{a} \oplus \bar{\dot{b}} = \bar{\dot{a}} \oplus \dot{b}. \quad (3)$$

由式(3)可知, 当输入、输出端口中取反个数为偶数个时, 对应表示式 $(\dot{a} \oplus \dot{b})$ 上有偶数个取反, 因此这些偶数个取反可以忽略而不会影响逻辑表达式的准确性. 即对于二输入“异或”门, 如 3 个端口取反的个数为偶数个时, 可以同时将这些取反操作忽略.

另外, 从式(3)可以得出, 当表示式 $(\dot{a} \oplus \dot{b})$ 中有一个变量取反时, 该取反符号可以在另一个变量和输出端移动而不会影响逻辑表达式的准确性. 考虑到任何一个奇数可以写成偶数加 1 的形式, 即当“异或”门 3 个端口取反的个数为奇数时, 等效为对任一端口的取反.

同理, 对于“同或”门时, 上述性质成立. 证毕.

图 3 显示了利用规则 1 和规则 2 实现复合门之间的等效变换. 图 3(a)为具有“AND-XOR”功能的逻辑复合门, 假设要将它等效转换为“OR-XNOR”功能的复合门. 利用规则 1, 将“AND”

门转化为“OR”门. 由于“AND”的 3 个端口都没有取反, 因此“OR”门的 3 个端口必须都取反, 这样才能保证“AND”门与“OR”门各个端口的取反个数之和为 3. 根据 $f(x, y) = x \oplus y = \bar{x} \odot y = x \odot \bar{y}$ 可得, 将“XOR”门转化为“XNOR”门只需在“XNOR”门的任何一个端口上增加一个取反就可以了. 于是就得到与图 3(a)逻辑功能等效的图 3(b). 图 3 中连线上的小圈表示该连线上传输的变量取反. 利用规则 2, 将图 3(b)中的一个输入的取反移到输出端, 得到图 3(c); 删除图 3(c)中同一条连线上 2 个取反运算, 就可得图 3(d). Algorithm 3 为目标复合门检测和等效变换算法伪代码.

Algorithm 3 match procedure

Input: node, faninnode, SHAREDTABLE,

destiststructure

Function: NodeIsComplex

if faninnode $\in$ SHAREDTABLE || node or faninnode is complex node || faninnode is terminal node then

return 0

else

if match(node, faninnode, destiststructure) then

return 1

end if

end if

Algorithm 3 是为了检测前后相连的两个基本逻辑门是否是目标复合结构, 或者是否可以转化成目标复合结构. 如果是, 则返回 1, 否则返回 0. 其中 SHAREDTABLE 和 destiststructure 的含义同 Algorithm 2, node 表示被检测的上一级节点, faninnode 表示被检测的下一级节点. 另外, 子程序 match() 用来检测待优化的逻辑网络的子结构, 或通过上面 2 个规则等效转换后的子结构是否与目标复合门的匹配.

3 实验结果及分析

本文的优化算法用 C 语言实现, 并用 MCNC 标准电路进行了测试. 考虑到电路内部的节点数与电路面积密切相关, 因此通常用节点数来衡量电路的面积^[13]. 表 2 为本文的方法与现有的逻辑综合工具的比较结果. 表 2 中第一列为 MCNC 标准

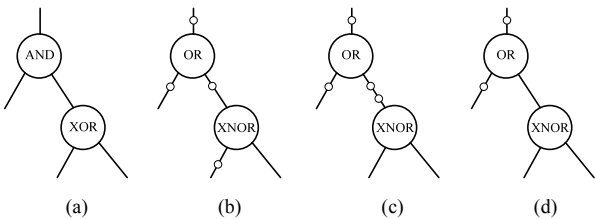


图 3 不同逻辑复合门之间的等效变换

电路名称, 而 ABC、BDS、CGMP 所在列中的数值表示采用不同方法后得到的构成电路内部节点数, 其中 ABC 表示用开源逻辑综合工具 ABC^[14]得到的结果, BDS 表示用 Yang 等^[12]提出的基于 BDD 拆分的逻辑综合工具 BDS 得到的结果, CGMP 表示用本研究方法得到的结果。

表 2 各类方法得到的结果比较

电路名	ABC	BDS	CGMP	$I_{ABC}/\%$	$I_{BDS}/\%$
5xp1	113	88	58	48.7	34.1
9sym	219	57	41	81.3	28.1
alu4	1 228	1 141	762	37.9	33.2
apex3	1 824	1 751	1 153	36.8	34.2
b12	67	64	45	32.8	29.7
clip	126	122	81	35.7	33.6
con1	17	21	14	17.6	33.3
cordic	67	43	36	46.3	16.3
duke2	509	623	506	0.6	18.8
e64	526	647	434	17.5	32.9
ex5p	586	408	322	45.1	21.1
misex1	55	74	50	9.1	32.4
misex3	1 229	903	634	48.4	29.8
o64	129	129	95	26.4	26.4
rd73	129	62	44	65.9	29.0
rd84	192	87	65	66.1	25.3
sao2	133	182	125	6.0	31.3
seq	1 836	2 522	1 690	8.0	33.0
squar5	48	49	34	29.1	30.6
t481	56	15	10	82.1	33.3
平均				37.0	29.3

表 2 中 I_{ABC} 和 I_{BDS} 分别表示用本研究方法得到的结果与 ABC 和 BDS 结果的比较。其中:

$$I_{ABC} = \frac{C_{ABC} - C_{CGMP}}{C_{ABC}} \times 100\%,$$

$$I_{BDS} = \frac{C_{BDS} - C_{CGMP}}{C_{BDS}} \times 100\%.$$

从表 2 可以得到, 经过逻辑复合门映射后, 电路节点数可得到有效的减少。从平均优化效果看, ABC 产生的节点数更多。因为 ABC 的优化采用与非图(Invert-AND Graph, AIG), 其优化结果极大地依赖电路的初始网络结构。本研究提出的 CGMP 算法的初始网络则基于 BDD 拆分, 因 BDD 在给定某种变量排序下是一种规范形表示, 所以如果一

个电路适合 BDD 拆分, 则再经我们的 CGMP 算法处理, 最终得到的网络会有更小的面积。以表 2 中电路 t481 为例, 该电路的逻辑表达式为:

$$v_{t481} = (((v_0 + \overline{v_1}) \odot (\overline{v_2} + v_3)) + ((v_4 + \overline{v_5}) \oplus (\overline{v_6} + v_7))) \odot (((v_8 + \overline{v_9}) \oplus (\overline{v_{10}} + v_{11})) + ((v_{12} + \overline{v_{13}}) \odot (\overline{v_{14}} + v_{15}))) . \quad (4)$$

电路 t481 用二输入门表示后, 包含 2 个 XOR 门、3 个 XNOR 门和 10 个 OR 门。根据复合门等效变换原理, 最终可在 t481 中构建 5 个 XNOR-OR 复合门, 因此结果可得到 10 个门的电路。但如用 ABC 工具所用的 AIG 来表示 XNOR 和 XOR 逻辑运算时, 与非图的结构将变得复杂, 导致 t481 的 ABC 优化结果不理想。同样, 如果一个电路不适合 BDD 拆分, 而适合 AIG 表示, 则 ABC 的优化结果将更好, 如电路 duke2 和 sao2。

由于我们提出的方法是采用了复合门的映射, 通过对逻辑网络的分析, 将部分逻辑网络构造为复合门并进行映射, 从而达到进一步减小了逻辑网络的节点数的目的。综合比较表明, 我们的方法比 BDS 减小约 29.3% 的节点数, 比 ABC 减小约 37.0% 的节点数。

4 结论

通过对逻辑网络一些结构用逻辑复合门进行替换实现了电路面积的优化。具体步骤包括: 基于 BDD 拆分的逻辑网络转换; 适合复合门实现的逻辑网络的搜索; 在搜索基础上确定复合门的种类, 即目标复合门。而对那些适合复合门实现, 但逻辑功能又不是目标复合门的电路结构, 则用复合门之间的逻辑等效转换, 转换为目标复合门。实验结果表明, 相比于 ABC 和 BDS 方法, 本研究算法能使电路的内部节点数量明显减少, 从而实现电路面积的优化。目前我们最终的复合门映射中仅使用了搜索结果中数量最多的复合门, 显然多种复合门在电路综合中的同时使用应该会取得更好的面积优化效果, 这有待于后续进一步研究。

参考文献:

- [1] REIS A I. Covering strategies for library free technology mapping[EB/OL]. [2015-06-16]. <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=803115&newsearch=true>

- &queryText=Covering%20strategies%20for%20library%20free%20technology%20mapping.
- [2] PULLERITS M, KABBANI A. Area minimization for library-free synthesis[EB/OL]. [2015-06-16]. <http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=5290465&url=http%3A%2F%2Fieeexplore.ieee.org%2Fstamp%2Fstamp.jsp%3Ftp%3D%26arnumber%3D5290465>.
- [3] MARQUES F S, ROSA L S, RIBAS R P, et al. DAG based library-free technology mapping[EB/OL]. [2015-06-16]. <http://dl.acm.org/citation.cfm?id=1228857&CFID=762315590&CFTOKEN=44866989>.
- [4] AMARU L, GAILLARDON P E, DE MICHELI G. MIXSyn: An efficient logic synthesis methodology for mixed XOR-AND/OR dominated circuits[EB/OL]. [2015-06-16]. <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6509585&queryText=MIXSyn:%20An%20efficient%20logic%20synthesis%20methodology%20for%20mixed%20XOR-%20AND%20FOR%20Dominated%20Circuits&newsearch=true>.
- [5] JIANG Y B, SAPATNEKAR S S, BAMJI C. Technology mapping for high-performance static cmos and pass transistor logic designs[J]. Very Large Scale Integration Systems, 2001, 9(5):577-589.
- [6] 黄春蕾, 王伦耀, 梁浩, 等. 三输入高性能 AND/XOR 复合门电路设计[J]. 浙江大学学报(理学版), 2015(3): 310-315.
- [7] 梁浩, 夏银水, 钱利波, 等. 低功耗三输入 AND/XOR 门的设计[J]. 计算机辅助设计与图形学学报, 2015(5): 940-945.
- [8] MO Y, ZHONG F, LIU H, et al. Efficient ordering heuristics in binary decision diagram-based fault tree analysis[J]. Quality and Reliability Engineering International, 2013, 29(3):307-315.
- [9] BRAND D. The best of ICCAD[M]. New York: Springer, 2003:65-72.
- [10] AMARU L, GAILLARDON P E, DE MICHELI G. Biconditional BDD: A novel canonical BDD for logic synthesis targeting XOR-rich circuits[EB/OL]. [2015-06-16]. <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6513657&newsearch=true&queryText=Biconditional%20BDD:%20A%20novel%20canonical%20BDD%20for%20logic%20synthesis%20targeting%20XOR-rich%20circuits>.
- [11] SUN F, XIA Y. BDD based detection algorithm for XOR-type logic[EB/OL]. [2015-06-16]. <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=4716264&queryText=BDD%20based%20detection%20algorithm%20for%20XOR-Type%20logic&newsearch=true>.
- [12] YANG C, CIESIELSKI M. BDS: A BDD-based logic optimization system[J]. Computer-aided Design of Integrated Circuits and Systems, 2002, 21(7):866-876.
- [13] CHEN Y C, WANG C Y. Logic restructuring using node addition and removal[J]. Computer-aided Design of Integrated Circuits and Systems, 2012, 31(2):260-270.
- [14] Berkeley Logic Synthesis and Verification Group. ABC: A system for sequential synthesis and verification[EB/OL]. [2015-06-16]. <http://www.eecs.berkeley.edu/~alanmi/abc/abc.htm>.

Area Optimization Based on the Complex Logic Gates Mapping

CEN Xu-meng, WANG Lun-yao*, XIA Ying-shui

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

Abstract: In this paper, an area optimization method is proposed based on the complex gates mapping. In order to map the complex gates, a logic circuit is turned into a multi-level network which consists of 2-input basic logic gates using the BDD decomposition. Then algorithms are designed to include searching for the sub-networks that can be replaced by the suitable complex gates, and the equivalent conversion between two different complex gates is utilized. The proposed method is implemented in C and tested by MCNC benchmarks. The experimental results show that, compared to the ABC and BDS tools, the number of the nodes of the tested circuits can be reduced by about 35% and 29% respectively by applying the proposed method.

Key words: complex gate; mapping; BDD; logic optimization

(责任编辑 史小丽)