

基于矩形覆盖的 FPRM 电路多级面积优化

陈治文¹, 张会红^{1*}, 汪鹏君^{1,2}, 张跃军¹, 张晓伟¹

(1. 宁波大学 电路与系统研究所, 浙江 宁波 315211; 2. 温州大学 数理与电子信息工程学院, 浙江 温州 325035)

摘要: Reed-Muller(RM)逻辑因其优越的性能而广泛应用于数字电路, 本文提出一种固定极性 RM(Fixed Polarity RM, FPRM)逻辑函数的多级优化方法. 首先将电路表示成 XOR/AND 形式的 FPRM 逻辑函数, 再计算函数的 kernels 和 co-kernels, 并由其生成矩阵, 然后从矩阵中搜索尽可能多的矩形覆盖, 利用矩阵分块和贪心策略逐步提取公共变量, 最后生成多级逻辑表达式. MCNC Benchmarks 测试后的结果表明, 本方法得到的表达式比原 RM 逻辑表达式减少 66% 的文字(literals)数目, 比 onset 表方法的优化结果减少 19%.

关键词: 逻辑优化; Reed-Muller 函数; 多级; Kernels

中图分类号: TN41

文献标志码: A

文章编号: 1001-5132 (2019) 04-0032-06

RM 逻辑实现的算术电路、奇偶校验电路及通信电路等多种电路比传统布尔形式电路在面积、功耗方面有着明显的优势, 因此 RM 逻辑优化引起了研究者的广泛关注^[1-2]. XOR/AND 和 XNOR/OR 是 RM 逻辑函数的 2 种形式, 以 XOR/AND 形式为例, 根据其极性形式可分为 FPRM 逻辑函数和混合极性 RM(Mixed Polarity Reed-Muller, MPRM)逻辑函数.

二级网络最小化是 FPRM 和 MPRM 优化的重要环节, 通过搜索最优极性以满足相应电路性能的要求. 当逻辑函数从布尔形式向 FPRM 形式转换以及在不同极性 FPRM 间转换时, 列表法凭借其易于计算机实现、效率高等特点被广泛使用^[3]. 然而列表法的操作对象是最小项, 其处理效率与电路输入数目成指数相关, 因此难以处理规模较大的电路. 文献[4]提出将不相交项与列表法相结合的快速转换方法, 其算法运行时间对电路输入输出数目不敏感, 仅与乘积项个数有关. 一个 n 输入的电路有 2^n 个极性, 各极性下的展开式不尽相同, 电路性能相差甚远, 利用智能算法可以避免穷尽整个极性空间来搜索满足电路性能的最佳极性. 文献[5]提出基于乘积项共享的遗传算法来搜索最

佳极性. 文献[6]提出了一种混合极性列表技术, 结合格雷码实现了多输出布尔逻辑向 MPRM 逻辑的快速转换. 文献[7]针对 MPRM 最小化时过早收敛的问题, 提出了一种基于相异度的局部改善策略的混合遗传算法. 文献[8]提出了一种基于多策略离散粒子群优化的极性搜索方法. 以上这些二级网络的最小化方法减少了 RM 表达式的乘积项, 一定程度优化了 RM 逻辑电路的面积.

多级优化可以获取更好的面积、功耗等性能. Xia 等^[9]提出应用 onset 表来进行多级混合极性 RM (Multi-level Mixed Polarity Reed-Muller, MPMRM) 优化, 而 onset 表对原 FPRM 表达式中取“1”的位置进行记录和操作, 在合理的时间内得到相应的 MPMRM 表达式. 但是该方法仅适用于单输出电路, 且提取公共变量能力不强. 文献[10]通过记录和分析变量出现频率对 onset 表结构进行优化, 从而获得了更优的公共变量提取能力. Wang 等^[11]提出了多输出 MPMRM 优化, 该方法分析和计算了 onset 表中“1”变量的位置和频率, 同时使用 MUX (Multiplexer) 完成了多输出 FPRM 表达式的公共变量的搜索和提取. 然而, 应用 onset 表的方法后仍会有部分公共变量未能成功提取, 因 onset 表在提

收稿日期: 2019-01-21.

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(61871244, 61704094); 宁波市自然科学基金(2018A610039); 宁波大学研究生科研创新基金.

第一作者: 陈治文(1993-), 男, 安徽池州人, 在读硕士研究生, 主要研究方向: 集成电路逻辑优化. E-mail: 1184369543@qq.com

*通信作者: 张会红(1976-), 女, 河北定州人, 博士/副教授, 主要研究方向: 低功耗、高信息密度集成电路. E-mail: zhanghuihong@nbu.edu.cn

取时会不断拆分, 而拆分后的若干 onset 表又难以提取部分公共变量, 且 onset 表的大小与电路输入相关, 电路输入数目较大时, 优化效率不高. 近几年来, ESOP(Exclusive-Or Sum Of Products)表达式等类 RM 表达式的最小化问题引起了众多研究者的关注^[12-14], RM 逻辑展开式作为逻辑函数的一种规范展开式仍具有重要研究意义.

本文结合 kernels 概念^[15]提出一种 RM 逻辑函数的多级优化方法. 首先获得 FPRM 的代数表达式, 然后提取其 kernels 和 co-kernels, 并生成由这两者构成的矩阵, 再对矩阵进行公共变量的搜索和分析, 最后整理可得 MPMR 表达式. 经过多组 MCNC Benchmarks 验证, 这种提取公共变量的方法比 onset 方法更高效, 且对电路输入输出不敏感, 仅与 FPRM 表达式的乘积项数有关.

1 kernels 与 co-kernels

逻辑函数在极性 i 下的固定极性 XOR/AND 展开式如下:

$$f_i(\dot{x}_{n-1} \cdots \dot{x}_1 \dot{x}_0) = \bigoplus_{j=0}^{2^n-1} b_j \pi_j, \quad (1)$$

式中, $\bigoplus \Sigma$ 表示乘积项的异或和; π_j 表示各变量组成的乘积项; b_j 为二进制数, 取“0”时, 表示 π_j 项不在表达式中出现, 取“1”时, 表示 π_j 在表达式中出现. 在 FPRM 表达式中, 各变量只能以其原型或其补的一种形式出现, 不能同时出现. 当若干个乘积项存在公共变量时, 提取出公共变量后逻辑不变, 且简化了表达式结构. 例如, $F_{\text{FPRM}} = ab \oplus bc = b(a \oplus c)$, 其中, b 作为公共变量提取出来, 使得 a 、 c 异或后再和 b 相乘, 变换前后逻辑函数功能不变. 本文目的是合理高效地提取尽可能多的公共变量, 完成 RM 逻辑函数的多级优化. 为了方便分析, 先给出以下定义^[15-16], 其中 F 、 G 和 H 均是 FPRM 表达式(XOR/AND 形式).

定义 1 单变量原型或其补的形式称为 literal; 一个或多个 literal 的乘积称为 cube.

定义 2 若表达式中的任意 2 个 cube 都不存在包含或被包含的关系, 则称该表达式为代数表达式; 若代数表达式 F 、 G 和 H 满足 $F = G * H$ (“*”表示代数乘法), 则 G 是 F 的一个代数因子, 称作 F 能被 G 整除, 记 $F / G = H$.

定义 3 若一个表达式不能被任何其他表达式整除, 则称该表达式是 cube-free.

定义 4 若一个表达式 F 除以一个 cube c 后, 得到若干个 cube-free, 那么这些 cube-free 的集合称为 kernels, 记为 $K(F)$; 所有 cube c 的集合称为 co-kernels, 记为 $C(F)$.

定义 1 和 2 给出了 literal、cube、FPRM 表达式和整除的概念. 例如 $F_{\text{FPRM}} = abc \oplus bce$, 则 $F_{\text{FPRM}} / (ba) = a \oplus e$, 此时有 $G = bc$, $H = a \oplus e$. 由定义 3 可知, F_{FPRM} 、 G 、 $ab \oplus be$ 均不是 cube-free, H 是 cube-free. cube-free 的定义保证了公共变量提取的完备性, 即任意 cube-free 中的 cubes 无公共变量; 由定义 4 可得 $K(F_{\text{FPRM}}) = \{a, e\}$, $C(F_{\text{FPRM}}) = \{bc\}$, 并称 $(a, e) / (bc)$ 为一组 kernels/co-kernels.

2 公共变量的提取

布尔逻辑中, 多级网络优化的应用十分广泛, 一些经典逻辑综合工具如 Espresso、ABC 等均提供 SOP 表达式的公共变量提取方法. 由于 FPRM 表达式不同于 SOP 表达式, 经典有效的方法难以直接复用, 这就造成了目前关于 RM 逻辑的优化只停留在二级网络阶段. 而文献[9-11]方法均存在一定局限性, 从适用性、全面性出发, 本文将复杂的表达式计算转换成客观的图形化操作. 首先计算 FPRM 的 $K(F)$ 和 $C(F)$, 再根据 $K(F)$ 和 $C(F)$ 构建矩阵 M , 最后应用智能算法在矩阵中搜索尽可能多的矩形覆盖来完成公共变量的提取.

对于 FPRM 表达式 F , 首先求出 kernels 集合 $K(F)$ 和 co-kernels 集合 $C(F)$:

$$\begin{cases} K(F) = \{k_{1,1}, k_{1,2}, \dots, k_{i,j}, \dots, k_{p,n}\}, \\ C(F) = \{c_1, c_2, \dots, c_l, \dots, c_m\}, \end{cases} \quad (2)$$

式中, p 表示 F 的输出数目; m 表示 co-kernels 中 cubes 的数目; n 表示 kernels 中的 cubes 数目; c_l 表示 F 的 co-kernel; $k_{i,j}$ 表示当前 cube 属于第 i 个输出的 kernel, 且是第 j 个 kernel.

根据 $K(F)$ 和 $C(F)$, 计算矩阵 $M = K(F)^T \cdot C(F)$, 具体如(3)式所示.

对于矩阵 M , 搜索所有矩形覆盖(如(3)式中由方括号标出的部分), 矩形覆盖应满足以下条件:

- ①至少覆盖 2 列元素;
- ②每行、每列的元素数目应相同;

$$\mathbf{M} = \begin{matrix} & k_{1,1} & k_{1,2} & \cdots & k_{i,j} & k_{m,n} \\ \begin{matrix} F \\ F \\ F \\ F \\ F \end{matrix} & \begin{matrix} c_1 [c_1 \cdot k_{1,1} \ c_1 \cdot k_{1,2}] \cdots \\ c_2 \\ \cdots \\ c_i \\ c_m \end{matrix} & & & & \end{matrix} \cdot (3)$$

③存在 2 个覆盖的 $K(F)$ 和 $C(F)$ 对应且相同, 则可合并为一个覆盖。

条件①和②是矩形覆盖的构成条件, 条件③则保证了矩形覆盖的最大化。确定一个符合条件的覆盖后, 用(4)式计算当该覆盖应用于公共变量提取时, 可减少的 literals 数目 $V(F)$ 。当覆盖应用于公共变量提取时, 将该覆盖的列元素和提取为公因子, 并新增为节点 k_q , 再把 k_q 代入到原矩阵中更新, 继续搜索下一个覆盖。

$$\begin{cases} v_{ij} = (\text{矩形覆盖中cube } c_i \cdot k_{i,x} \text{ 的literals数目}), \\ w_i^c = (\text{cube } c_i \text{ 的literals数目})+1, \\ w_j^k = (\text{cube } k_{i,x} \text{ 的literals数目}), \\ V(F) = \sum_{i \leq m, j \leq n} v_{ij} - \sum_{i \leq m} w_i^c - \sum_{j \leq n} w_j^k. \end{cases} \quad (4)$$

由(4)式可计算出当前覆盖带来 literal 数目的减少量, 然而由于矩阵的覆盖划分方法、数量以及每个覆盖可减少的 literal 数目不尽相同, 因此, 搜索并提取最佳覆盖, 以使 literal 数目减少量达到最大值的问题属于 NP-Hard 问题。本文使用贪心算法求解该问题, 依次选择当前矩阵 $\mathbf{M}$ 下 $V(F)$ 值最大时的覆盖以进行公共变量的提取, 具体步骤如下:

(1)计算表达式 F 的 $K(F)$ 和 $C(F)$, 构建矩阵 $\mathbf{M}$, $s=0, V(F)=0$;

(2)计算矩阵 $\mathbf{M}$ 中的所有覆盖的 $V(F)$, $V(F)$ 最大时为 $V_{\max}$, $V_F = V_F + V_{\max}$;

(3)找出 $V_{\max}$ 对应的覆盖, 新增节点 $X_s = \oplus \Sigma k_{i,j}$, $k_{i,j}$ 为覆盖的列元素, 即 $k_{i,j} \in K(F)$;

(4)将 X_s 代入到原表达式 F 中, 更新矩阵 $\mathbf{M}$;

(5) $s=s+1$, 执行(2), 直至所有覆盖提取完成。

以上是单输出表达式 F 应用矩形覆盖的方法, 在合理时间内最大程度地提取公共变量, 提取的公共变量以新增节点的方式加入原表达式中。矩形覆盖元素的实质是 FPRM 表达式中含公共因子的 cubes, 当多个 cubes 排成一列(一行)时, 其另一项乘积因子 kernels(co-kernels)必然相同, 可作为公

共变量提取。同样, 对于多输出电路也适用。因为上述方法将每个输出都看成一个节点, 提取到的公共变量仍然以节点方式新增, 这对电路的输出数目没有限制。需要注意的是, 矩阵 $\mathbf{M}$ 中作为行元素的 c_i 属于不同输出(节点)的 co-kernels, 因此 c_i 与 c_j 可能相同, 但相同输出(节点)的 c_i 必定是唯一的; 而矩阵 $\mathbf{M}$ 中作为列元素的 $k_{i,j}$ 则不对输出进行区分, 即该元素是一个或多个输出表达式中若干 cubes 的公共变量。

3 FPRM 表达式的多级优化方法

基于 XOR/AND 形式的固定极性 RM 逻辑表达式如(1)式所示, 对(1)式的多级逻辑优化时, 已有研究通常对表达式的真值表或 onset 表进行拆分和合并, 计算量大且复杂、且优化不彻底。据此, 本文基于多输出 FPRM 表达式提出一种基于矩形覆盖的多级优化方法, 以下即为具体优化方案。

3.1 计算 FPRM 表达式的 kernels 和 co-kernels

当表达式中 cubes 较多时, 直接计算所有 cubes 的 kernels 较为困难, 此时, 先计算其中若干 cubes 的 kernels 和 co-kernels, 再将 co-kernels 中所有 cubes 逐一整除式中其他 cubes, 该 co-kernels 则是所有能够被整除 cubes 的公共变量, 同时计算这些 cubes 的 kernels; 然后计算另外 2 个 cubes, 直到所有 cubes 计算完毕; 最后将所有的 kernels 存入集合 K , 将所有的 co-kernels 存入集合 C 中。

3.2 矩形覆盖的搜索和提取

构建矩阵 $\mathbf{M}$, 并将所有 $k_i (k_i \in K)$ 和 $c_i (c_i \in C)$ 相乘结果属于原表达式乘积项的 cubes 填入矩阵中。搜索 $\mathbf{M}$ 中满足条件的矩形覆盖, 并以(4)式中 $V(F)$ 最大时的覆盖为最优覆盖, 逐一进行公共变量的提取。实验时发现, 因为每次新增节点后, 需重新计算 C 和 K , 然后构建 $\mathbf{M}$, 因此更新 $\mathbf{M}$ 耗时较多。对此, 提出矩阵分块计算方法:

(1)从矩阵 $\mathbf{M}$ 中搜索最优覆盖, 然后以紧贴该覆盖一边的元素为界限, 划分出规模较均匀的两个矩阵 $\mathbf{M}_1$ 和 $\mathbf{M}_2$ (图 1)。

(2)对矩阵 $\mathbf{M}_1$ 和 $\mathbf{M}_2$ 重复上述步骤(1), 直到该矩阵内的元素不大于预先设置的界限常数 R 。

(3)对这些划分的小规模矩阵分别进行多级优化, 得到多个不完全的多级表达式。以单输出为例,

		$k_{1,1}$	$k_{1,2}$	$\cdots$	$k_{i,j}$	$k_{m,n}$
$M =$	$F \quad c_1$	$[c_1 \cdot k_{1,1} \quad c_1 \cdot k_{1,2}]$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
	$F \quad c_2$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
	$F \quad \cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
	$F \quad c_i$	$\cdots$	$\cdots$	$\cdots$	$c_i \cdot k_{i,j}$	$\cdots$
	$F \quad c_m$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$c_m \cdot k_{m,n}$
		M_1		M_2	$\cdots$	

图 1 矩阵分块

得到的多级表达式为 $F = \oplus \Sigma c_i$, 其中, c_i 是多级表达式.

(4) 将 F 视为广义的 FPRM 表达式进行多级逻辑优化. 比如, $c_i = (a \oplus b)(c \oplus d)ef$, 将 $a \oplus b$ 、 $c \oplus d$ 、 ef 视为单个变量, 则 c_i 仍然是乘积项, F 则是广义上的 FPRM 表达式.

3.3 变量取反

多级表达式 F 中往往存在常量“1”, 需要将“1”相异或的变量取反, 有利于进一步优化.

举例说明 FPRM 表达式的多级优化方法: 读入电路文件后, 得到多输出 FPRM 表达式如(5)式, 由此计算的 kernels/co-kernels 如图 1 所示.

$$\begin{cases} F = af \oplus bf \oplus ag \oplus cg \oplus ade \oplus bde, \\ G = af \oplus bf \oplus ace \oplus bce, \\ H = ade \oplus cde. \end{cases} \quad (5)$$

由图 2 可得, $K = \{a, b, c, ce, de, f, g\}$, $C = \{a_F, b_F, c_F, de_F, f_F, g_F, a_G, b_G, ce_G, f_G, de_H\}$ (下标表示该 cube 所属的输出), 根据集合 K 和 C 构建矩阵 M (图 3). 搜索最优覆盖时, 图 3 中方框部分合并成最优覆盖, 根据(4)式计算可得 $V_{\max} = 20 - 10 - 2 = 8$.

新增结点 $X_1 = a \oplus b$, 提取公共变量如下:

$$\begin{cases} F = (a \oplus b)f \oplus ag \oplus cg \oplus (a \oplus b)de, \\ G = (a \oplus b)f \oplus (a \oplus b)ce, \\ H = ade \oplus cde. \end{cases} \quad (6)$$

验证: $V_{(5)} - V_{(6)} = 30 - 22 = 8 = V_{\max}$, 证毕.

接下来将 X_1 视为新的输出, 计算新的集合 K 和 C , 更新矩阵 M 后, 依次搜索其他矩形覆盖, 并进行公共变量的提取, 方法与上述一致. 以上多级 FPRM 优化方法流程如图 4 所示.

4 实验结果和分析

提出的基于矩形覆盖的 FPRM 逻辑函数的多级优化方案已经通过 C 语言编程实现, 并在 3.30 GHz CPU 频率, 4 GB 内存, Windows 10 操作系统下的 DELL PC 机上使用 Visual Studio 2015 编译运行.

输出	kernels	co-kernels
F	$(de \oplus f \oplus g)$	a
F	$(de \oplus f)$	b
F	$(a \oplus b)$	de, f
F	g	c
F	$(a \oplus c)$	g
G	$(ce \oplus f)$	a, b
G	$(a \oplus b)$	f, ce
H	$(a \oplus c)$	de

图 2 kernels/co-kernels 计算

		a	b	c	ce	de	f	g
$M =$	$F \quad a$					ade	af	ag
	$F \quad b$					bde	bf	
	$F \quad c$							cg
	$F \quad de$	ade	bde					
	$F \quad f$	af	bf					
	$F \quad g$	ag		cg				
	$G \quad a$				ace		af	
	$G \quad b$				bce		bf	
	$G \quad ce$	ace	bce					
	$G \quad f$	af	bf					
	$H \quad de$	ade		cde				

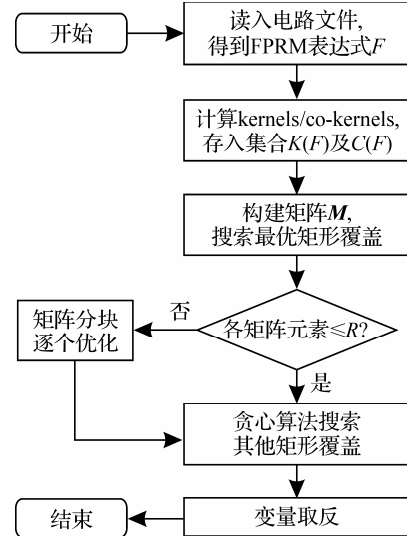
图 3 矩阵 M 和矩形覆盖

图 4 FPRM 表达式的多级优化流程

表 1 是本方法与极性优化方法、onset 表方法经过 18 个 MCNC benchmarks 和 3 个随机生成电路测试后的实验结果对比. 电路面积以 literals 数目衡量, 而 literals 数是一种粗略描述电路面积的方法, 在文献[11-13]中也被应用, 鉴于本文方法与其他对照方法的电路结构相似, 因此以 literals 计算面积有较强说服力. 表 1 中, 前 2 列分别是电路名、输入输出数目; 第 3 大列是各方法所得的 literals 数目, 这些方法分别是采用不相交乘积项获得的最佳极

性下多输出 FPRM 表达式、采用 onset 表进行优化得到的 MMPRM 表达式和本文方法; “ I_{imp1} ” “ I_{imp2} ” 表示后两者方法分别对于第 1 种方法的改进 literals 数目的百分比, 计算方式如下:

$$I_{imp_i} = (L_x - L) \div L_x \times 100\%, \quad (7)$$

式中, $i=1, 2$; L_x 为不相交项方法、onset 表方法优化后的 literals 数, L 为本方法优化后的 literals 数. CPU₁ 和 CPU₂ 分别是 onset 表方法、本文方法处理的所需的 CPU 时间, “<0.01” 表示程序运行完屏幕显示的时间为 0. 界限常数 R 设为 100.

从表 1 中可见: (1)所提方法对于 FPRM 表达式的多级优化效果显著. 所有测试电路平均减少的 literals 数目约为 66%, 个别电路如 “t481” “alu4” 等减少量在 80%以上; (2)所提方法比 onset 表方法的处理结果更优. 前者比后者的 literals 减少量高出约 19%, 尤其是 3 组随机生成的 FPRM 表达式, 对比效果十分明显. 这是因为 onset 方法是对 onset 表多次拆分后进行公共变量提取, 最后将各部分所得结果合并, 其表格性质决定了无法对合并结果进行“二次优化”; 而本文方法在处理较大规模

电路时, 分块既能带来处理效率的上升, 也不影响后续的重复优化; (3) 2 种方法对于输入电路的规模均不太敏感. 不过 onset 方法对于规模较大的电路如 “TEST2” “TEST3” 存在优化慢、效率低的缺点, 所提方法的处理时间仅与乘积项数有关, 与输入输出数无关.

5 结论

针对固定极性 XOR/AND 表达式的多级优化问题, 提出了一种基于矩形覆盖的公共变量提取方案. 该方法计算表达式的 kernels/co-kernels 后, 将其填入矩阵中, 对于规模较大矩阵, 采用分块和贪心策略搜索矩阵中满足条件的矩形覆盖, 并逐步完成公共变量的提取, 规模较小电路直接采用后者策略, 矩阵分块有利于减小算法处理时间, 贪心策略保证了结果正确性. 经 MCNC benchmarks 测试表明, 所得结果比已有研究的结果更优且更全面, 效率更高. 本文方法还具有对电路输入大小不敏感、适用于多输出电路等优点, 后续计划将本方法扩展至 ESOP 表达式的多级优化.

表 1 多输出 FPRM 表达式优化结果对比

电路名	i/o	literals 数目			$I_{imp1}/\%$	$I_{imp2}/\%$	CPU ₁	CPU ₂
		P_best	onset	proposed				
9sym	9/1	964	415	231	76.0	44.3	<0.01	<0.01
newill	8/1	52	31	24	53.8	22.6	<0.01	<0.01
newtag	8/1	46	26	24	47.8	7.7	<0.01	<0.01
life	9/1	512	189	166	67.6	12.2	<0.01	<0.01
sym10	10/1	2 067	847	496	76.0	41.4	0.08	0.02
ryy6	16/1	1 386	513	295	78.7	42.5	<0.01	<0.01
t481	16/1	14 436	10 105	2 716	81.2	73.1	5.75	12.05
sao2	10/4	799	256	314	60.7	-22.7	<0.01	<0.01
rd84	8/4	753	369	274	63.6	25.7	<0.01	<0.01
rd73	7/3	441	246	196	55.6	20.3	<0.01	<0.01
clip	9/5	2 388	1 026	788	67.0	23.2	<0.01	<0.01
b12	15/9	1 130	497	299	73.5	39.8	<0.01	<0.01
misex1	8/7	301	178	139	53.8	21.9	<0.01	<0.01
inc	7/9	673	397	366	45.6	7.8	<0.01	<0.01
bw	5/28	555	480	457	17.7	4.8	<0.01	<0.01
apex4	9/19	15 205	10 795	7 079	53.4	34.4	8.03	0.08
alu4	14/8	52 674	15 275	9 423	82.1	38.3	10.23	7.78
misex3	14/14	68 448	21 903	17 178	74.9	21.6	23.50	18.23
TEST1	20/15	80 275	48 065	14 980	81.3	68.8	49.72	19.41
TEST2	25/20	152 788	93 200	21 063	86.2	77.4	135.00	37.78
TEST3	30/25	50 008	38 764	8 139	83.7	79.0	38.53	1.24
Avg	/				65.7	32.6	11.09	4.60

参考文献:

- [1] 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.
- [2] He Z X, Xiao L M, Gu F, et al. An efficient and fast polarity optimization approach for mixed polarity Reed-Muller logic circuits[J]. Frontiers of Computer Science, 2017, 11(4):728-742.
- [3] Al Jassani B A, Urquhart N, Almaini A E A. Manipulation and optimisation techniques for Boolean logic[J]. IET Computers & Digital Techniques. 2010, 4(3):227-239.
- [4] 王玉花, 王伦耀, 夏银水. 基于不相交项并行列表技术的 FPRM 实现[J]. 电子与信息学报, 2014, 36(9):2258-2264.
- [5] Apangshu D, Sambhu N P. Thermal aware FPRM based AND-XOR network synthesis of logic circuits[C]. IEEE 2nd International Conference on Recent Trends in Information Systems, 2015:497-502.
- [6] 李辉, 汪鹏君, 王振海. 混合极性列表技术及其在 MPRM 电路面积优化中的应用[J]. 计算机辅助设计与图形学学报, 2011, 32(3):527-533.
- [7] 卜登立. 基于混合遗传算法的 MPRM 最小化[J]. 浙江大学学报(理学版), 2016, 43(2):184-189.
- [8] 符强, 汪鹏君, 童楠, 等. 基于多策略离散粒子群算法的 MPRM 电路延时与面积优化[J]. 电子学报, 2016, 44(5):1202-1207.
- [9] Xia Y S, Wang L Y, Meng Y, et al. Onset based optimization of multi-level mixed polarity Reed-Muller Functions[C]. Proceedings of the 5th WSEAS International Conference on Instrumentation, Measurement, Circuits and Systems, 2006:246-251.
- [10] Wang L Y, Xia Y S. A fast algorithm for multi-level mixed-polarity Reed-Muller functions optimization[C]. 11th IEEE International Conference on Communication Technology, 2008:509-512.
- [11] Wang L Y, Xia Y S, Chen X X, et al. Reed-Muller function optimization techniques with onset table[J]. Journal of Zhejiang University Science C: Computer & Electronics, 2011, 12(4):288-296.
- [12] Papakonstantinou G. A parallel algorithm for minimizing ESOP expressions[J]. Journal of Circuits Systems & Computers, 2014, 23(1):72-85.
- [13] Papakonstantinou G. Exclusive or sum of complex terms expressions minimization[J]. Integration, the VLSI Journal, 2017, 56:44-52.
- [14] Zhang Q W, Wang P J, Hu J, et al. Cube-based synthesis of ESOPs for large functions[J]. Chinese Journal of Electronics, 2018, 27(3):527-534.
- [15] Brayton R K, Redull R, Vincentelli A S. MIS: A multiple-level logic optimization system[J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 1987, 6(6):1062-1081.
- [16] Brayton R K, Hachtel G, Vincentelli A S. Multilevel logic synthesis[J]. Proceedings of the IEEE, 1990, 78(2): 264-300.

Rectangular overlays based multi-level area optimization of FPRM circuit

CHEN Zhiwen¹, ZHANG Huihong^{1*}, WANG Pengjun^{1,2}, ZHANG Yuejun¹, ZHANG Xiaowei¹

(1.Institute of Circuits and Systems, Ningbo University, Ningbo 315211, China;

2.College of Mathematics, Physics and Electronic Information Engineering, Wenzhou University, Wenzhou 325035, China)

Abstract: Reed-Muller (RM) logic is widely used in digital circuits for its superior performance. A multi-level optimization method for Fixed Polarity RM (FPRM) logic function is proposed in this paper. First, logic circuits are converted into FPRM logic function in XOR/AND form. Next, the kernels and co-kernels of the function are calculated to generate a matrix. Then rectangular overlays are searched for in the matrix, in which process the matrix partition and greedy strategy are used to gradually extract the public variables. Finally, the multi-level logic expressions are generated. The results tested using the MCNC Benchmarks show that the number of literals obtained by the proposed method is 66% less than that of polarity optimization, and 19% less than that of the onset table method.

Key words: logic optimization; Reed-Muller function; multi-level; Kernels

(责任编辑 章践立)