

基于不相交项并列表技术的 FPRM 实现

王玉花 王伦耀* 夏银水

(宁波大学信息科学与工程学院 宁波 315211)

摘要: 针对传统列表技术在逻辑函数从 AND/OR 形式转化成固定极性 Reed-Muller(FPRM)过程中只能处理小规模电路的不足, 该文提出一种基于不相交乘积项的并列表技术。该技术能有效避免转化算法因逻辑函数输入变量增加引起最小项数量激增而导致效率低下甚至无法工作这种情况。另外, 不同于已发表的用于实现大电路的转化算法, 待处理的电路结构对该方法的性能影响很小。提出的算法用 C 语言编程实现, 并用 MCNC 标准电路进行测试。实验结果表明所提算法可以对更大规模电路实现快速 FPRM 转换, 并且算法速度对电路输入个数不敏感, 但与待处理逻辑函数的不相交乘积项的数量有关。

关键词: 数字电路; Reed-Muller (RM)逻辑; 固定极性; 并列表技术; 逻辑优化

中图分类号: TP331.2; TP391.72

文献标识码: A

文章编号: 1009-5896(2014)09-2258-07

DOI: 10.3724/SP.J.1146.2013.01536

FPRM Conversion Using Parallel Tabular Technique with Disjointed Products

Wang Yu-hua Wang Lun-yao Xia Yin-shui

(School of Information Science and Engineering, Ningbo University, Ningbo 315211, China)

Abstract: With the deficiency of the published tabular techniques based algorithms which can only handle small functions in the conversion from AND/OR forms to the Fixed Polarity Reed-Mull (FPRM) forms, a novel parallel tabular technique using the disjoint products is proposed. By utilizing the disjointed products, the proposed algorithm is able to avoid the rapid increase of the minterms which leads the reported tabular technique based algorithms cannot run efficiently or even out of work. Furthermore, unlike the published algorithm for the large functions conversion, the circuit structure in progress has little effect on the performance of the proposed algorithm. The proposed algorithm is implemented in C language and tested under MCNC benchmarks. The experimental results show that the proposed algorithm can finish the polarity conversion fast for the larger circuits and the speed of the algorithm does not depend on the number of inputs of the circuits but the number of the disjointed products.

Key words: Digital circuit; Reed-Muller (RM) logic; Fixed polarity; Parallel tabular technique; Logic optimization

1 引言

在逻辑函数从常见的 AND/OR 形式向固定极性 Reed-Muller (Fixed-Polarity Reed-Muller, FPRM)展开过程中, 列表技术因便于计算机实现, 且有较高的效率而经常被采用^[1-4]。现有的采用列表技术的 FPRM 展开算法都基于逻辑函数的最小项。即在采用列表技术前, 先把逻辑函数的乘积项转换为最小项。由于逻辑函数的最小项的数量与 2^n 成正相关, 其中 n 为逻辑函数的输入变量个数, 因此, 随着输入变量的增加, 算法所要处理的数据量成2的指数级增加, 从而导致算法效率减低, 甚至

无法工作。这也是目前已经发表的列表技术只能做到对输入变量30个以内的逻辑函数实现FPRM的转化的主要原因。

在用列表技术将逻辑函数从 AND/OR 形式向 FPRM 表达式转化的研究中, 文献[5]针对混合极性的最佳极性优化的问题, 提出了一种基于并行表格技术的遗传算法, 克服了传统表格技术中因顺序产生相关项造成数据相关性问题的, 但是该方法处理的电路输入个数在20以内。文献[6]针对逻辑函数 AND/OR 形式展开式中无关项取舍不确定的特点, 结合并列表技术, 提出一种包含无关项逻辑函数的固定极性转换方法。但是文献[6]所给出的测试电路最大输入变量数为15。文献[7]提出一种专门针对大规模布尔函数的快速转换算法, 但该方法只能处理电路冗余度(即乘积项中很多变量没有出现)很大

2013-10-10 收到, 2014-02-18 改回

国家自然科学基金(61131001, 61228105), 浙江省自然科学基金(LY12F01014)和宁波市自然科学基金(2013A610009)资助课题

*通信作者: 王伦耀 wanglunyao@nbu.edu.cn

且多输出函数每行只能有一个‘1’的大电路，所以这种方法只适用于电路结构比较特殊的电路，算法的通用性有待提高。

Reed-Muller(RM)逻辑除固定极性展开外，也可以用混合极性(Mixed-Polarity Reed-Muller, MPRM)进行展开。文献[8]提出了一种混合多值离散粒子群优化算法实现 MPRM 逻辑综合；文献[9]提出了采用多数覆盖实现 MPRM 逻辑综合；文献[10]提出了一种混合极性下 RM 逻辑的功耗优化方法。此外，以 MPRM 子表达式之积的形式，即 SPP 形式，实现电路面积的优化的方法也被提了出来^[11,12]。近来，有些学者提出了在逻辑综合中同时采用 AND/OR 形式的布尔逻辑和 AND/XOR 形式的 RM 逻辑来实现电路的面积优化，即双逻辑综合^[13-15]。但无论是混合极性 RM 综合还是双逻辑综合，固定极性 RM 逻辑展开方法作为一种基本的 RM 逻辑优化方法而一直受到重视。

本文结合不相交乘积项和传统的列表技术，实现由 Boolean 函数乘积项表达式到固定极性的 AND/XOR 表达式的快速转换。最后与文献[7]的结果进行了对比，以检验提出的算法的效果。实验证明本文方法对测试电路的结构没有要求，不论处理小规模电路还是大规模电路，算法的速度都比较快，并且算法的运算速度对电路的输入个数不敏感，本文的方法将为基于最优极性搜索的 Reed-Muller 逻辑综合奠定基础。

2 基本概念

逻辑函数 f 可以表示为如式(1)所示的乘积项之和(Sum Of Product, SOP)形式。

$$f(x_0, x_1, \dots, x_{n-1}) = \sum_{i=0}^{k-1} p_i \quad (1)$$

式(1)中，“ \sum ”是逻辑“或”运算符， p_i 示第 i 个乘积项。对于构成 SOP 形式的逻辑项，其包含的变量形式可以是原变量，反变量和变量不出现 3 种形式。本文分别用“1”，“0”和“-”分别表示对应的变量为原变量，反变量和变量不出现。

另外，逻辑函数 f 也可以表示成式(2)所示的 Reed-Muller(RM)展开形式。

$$f(x_0, x_1, \dots, x_{n-1}) = \oplus \sum_{i=0}^{v-1} p_i \quad (2)$$

其中，符号“ $\oplus \sum$ ”表示乘积项之间是逻辑“异或”关系。

在 RM 逻辑函数中，变量可以以原变量形式出现，也可以以反变量形式出现。变量的这两种形式可以利用式(3)来实现。

$$\bar{x} \triangleq \delta_i \oplus x_i = \begin{cases} x_i, & \text{若 } \delta_i = 0, \text{ 称 } x_i \text{ 取正极性} \\ \bar{x}_i, & \text{若 } \delta_i = 1, \text{ 称 } x_i \text{ 取负极性} \end{cases} \quad (3)$$

显然，对于 RM 逻辑函数 f ，利用式(3)可以使 f 的任何一个变量 x_i ，在任何一个乘积项中，它的取值形式保持一致。即所有乘积项中的 x_i 的取值不能同时既有原变量又有反变量。RM 逻辑函数的变量按照某种取值形式进行展开得到的 RM 函数称为固定极性下的 RM 函数(Fixed Polarity Reed-Muller, FPRM)。 f 的极性值用 P 表示，其中 P 等于各个 δ_i 构成的二进制的值。

现以二变量 RM 逻辑函数为例说明不同极性的 RM 表达式复杂性的影响。

$$\begin{aligned} f_r(x_1, x_2, x_3) &= x_1 \oplus x_3 \oplus x_2 x_3 \oplus x_1 x_2 x_3 : (P = 0 = (000)_2) \\ &= \bar{x}_2 x_3 \oplus x_1 \oplus x_1 x_3 \oplus x_1 \bar{x}_2 x_3 : (P = 2 = (010)_2) \\ &= 1 \oplus x_3 \oplus \bar{x}_1 \oplus \bar{x}_1 x_2 x_3 : (P = 4 = (100)_2) \\ &= \bar{x}_3 \oplus \bar{x}_1 \oplus \bar{x}_1 x_2 \oplus \bar{x}_1 x_2 \bar{x}_3 : (P = 5 = (101)_2) \\ &= \bar{x}_3 \oplus \bar{x}_1 \bar{x}_3 \oplus \bar{x}_1 \bar{x}_2 \oplus \bar{x}_1 \bar{x}_2 \bar{x}_3 : (P = 7 = (111)_2) \end{aligned} \quad (4)$$

其中， P 表示极性情况。如 $P = 4 = (100)_2$ 表示极性为 4。(100)₂ 表示 4 的二进制表示，对应 $\delta_1 = 1, \delta_2 = 0, \delta_3 = 0$ 。由式(3)得到 $P = 4 = (100)_2$ 情况下，变量 x_1 以反变量形式出现，变量 x_2 和 x_3 以原变量形式出现。从式(4)可知，不同的极性对 RM 逻辑函数的表达式影响很大，显然当 RM 逻辑函数 f 的极性 P 取某个(或某些)值时， f 的表达式就会最简，此时的 P 就是 f 的最佳极性。因此，RM 逻辑函数的简化可以通过改变逻辑函数的极性来实现。

3 不相交乘积项的并行列表技术

3.1 基于最小项的列表技术

已发表的用于实现 AND/OR 形式的逻辑表达式向固定极性的 AND/XOR 逻辑表达式转换的列表技术都是基于最小项实现的。具体又可以分为串行列表技术和并行列表技术。串行列表技术一般由如下 5 个步骤组成。

步骤 1 将乘积项转换为最小项，删除重复项，并以二进制形式列出所有最小项；

步骤 2 用二进制表示待转化的极性，得到极性表达式。将所有最小项与所求极性表达式进行异或操作，得到新的逻辑项；

步骤 3 检测用二进制形式表示每个最小项中各个位上取值为“0”的情况，如果最小项 m_i 的第 j 位为 0，则产生新的项 m_i' 就是通过将 m_i 的第 j 位用 1 代替，并复制 m_i 其余所有的位得到；

步骤 4 检查所有最小项，删除包括新项在内

的成对出现的逻辑项；

步骤 5 重复步骤 3 和步骤 4 直到遍历完所有变量。剩余的未被删除的项即为所求的构成 FPRM 表达式的逻辑项。

串行列表技术的特点是转换过程需要对各个变量进行依次处理，每次循环只能对一位变量进行操作，极性转换的效率较低，尤其是随着电路变量数量的增多，导致最小项数量的以 2 的指数级增加，庞大的最小项数量使得极性转换时间急剧增加，进而无法实现对大电路的 FPRM 转换。

并行列表技术采用对逻辑项中取值为“0”的位进行并行方式处理。具体步骤为：

步骤 1 将乘积项转换为最小项，删除重复项，将剩余最小项以二进制形式列出；

步骤 2 用二进制表示待转化的极性，得到极性表达式，将所有最小项与所求极性表达式进行异或操作，得到新项；

步骤 3 设某一个新项有 s 个二进制位为 0，则对这 s 位进行展开，除原乘积项外，共可以产生 $2^s - 1$ 个新项；

步骤 4 重复步骤 3 直至操作完所有新项；

步骤 5 删除数量为偶数个的最小项，剩下的项即为对应极性的 AND/XOR 项。

例 1 对函数 $f(x_0, x_1, x_2) = x_1 \bar{x}_2 + x_0 \bar{x}_1$ 分别用串行列表技术和并行列表技术进行两种逻辑之间的转换。其中极性 $P = (001)_2$ 。

由表 1 得到采用串行列表技术后 f 的零极性下 RM 表达式为： $f(x_0, x_1, x_2) = \oplus \sum(2, 3, 4, 6)$ 。

由表 2 得用并行列表技术后 f 的零极性下 RM 表达式为： $f(x_0, x_1, x_2) = \oplus \sum(2, 3, 4, 6)$ 。

表 1 串行表格技术转换结果

乘积项	最小项	要求极性	新最小项	新项		
				x_0	x_1	x_2
-11	111		110	110	111	111
10-	011	001	010		110	011
	100		101		101	101
	101		100			

表 2 并行表格技术转换结果

乘积项	最小项	要求极性	新最小项	新项		
				x_0	x_1	x_2
-11	111		110		111	111
10-	011	001	010	110	011	111
	100		101		111	111
	101		100	110	101	111

由例 1 可知，串行表格技术和并行表格技术所得的结果一样，但从已发表的研究来看^[15]，并行表格技术转换的速度更快。

3.2 基于不相交乘积项的列表技术

由于已发表的用列表技术实现逻辑函数 FPRM 转换时，先将待转换逻辑函数化为最小项之和形式，然后再实行 FPRM 转换。因此当逻辑函数的输入变量比较多时，比如输入变量数大于 30，其对应的最小项数可能数以亿计。庞大的最小项数量给计算机内存和算法的运算速度带来很大影响，这也是目前已发表的基于列表技术进行 FPRM 转换时往往只能处理输入变量小于 30 的逻辑函数的主要原因。而相比于最小项，一个逻辑函数的不相交项的数量往往远远小于它的最小项个数；并且与最小项类似，不相交项之间也不存在公共部分。因此，以不相交项为基础，根据列表法产生的新的乘积项的过程中，不会因为原始乘积项之间存在公共部分而产生重复的新的乘积项，进而导致最终转换结果的错误。本文提出的基于不相交乘积项的列表技术是在下面定理的基础上提出的。

定理 1 逻辑项在固定 p 极性预处理中，取值为“-”的变量保持不变。即 $(-\oplus 1 = -)$, $(-\oplus 0 = -)$ 。

证明 用归纳法证明。设待预处理的 k 变量逻辑项 $p = (x_1 x_2 \cdots x_k)$ ，其中 p 中有 n 个变量取值为“-”， $n < k$ 。

(1) 当 $n = 1$ 时。假设逻辑项只在第 i 位的变量取值为“-”，即

$$p = \{p_{i-1}, -, p_{i+1}\} \tag{5}$$

其中， p_{i-1} 表示 p 中前 $i-1$ 个变量构成的表达式。 p_{i+1} 表示从第 $i+1$ 到 k 个变量构成的表达式， $i \geq 1$ 。极性表达式 H 的第 i 位的值为“1”，即 $H = \{H_{i-1}, 1, H_{i+1}\}$ 。式中的 H_{i-1}, H_{i+1} 的含义同式(5)。

由于 $p = \{p_{i-1}, -, p_{i+1}\}$ 可以展开的 2 个逻辑项分别为 $p_1 = \{p_{i-1}, 0, p_{i+1}\}$ 和 $p_2 = \{p_{i-1}, 1, p_{i+1}\}$ 。 p_1, p_2 与 H 的位“异或”得到 $p'_1 = \{p'_{i-1}, 1, p'_{i+1}\}$ 和 $p'_2 = \{p'_{i-1}, 0, p'_{i+1}\}$ 。合并 p'_1 和 p'_2 ，最终得到 $p' = \{p'_{i-1}, -, p'_{i+1}\}$ ，即 p 的第 i 位的“-”保持不变。同理，当极性表达式为 $H = \{H_{i-1}, 0, H_{i+1}\}$ 时，得到的 p'_1 和 p'_2 分别为 $p'_1 = \{p'_{i-1}, 1, p'_{i+1}\}$ 和 $p'_2 = \{p'_{i-1}, 0, p'_{i+1}\}$ ，最终得到 $p'' = \{p'_{i-1}, 1, p'_{i+1}\}$ ，即第 i 位的“-”保持不变。

(2) 假设当 $n = m, 1 \leq m < k$ 时定理 1 成立。即当 $p = \{(-, \cdots, -)_m, p_{m+1}\}$ ， $H = \{H_m, H_{m+1}\}$ 时，有 $p' = \{(-, \cdots, -)_m, p_{m+1}\}$ 。其中 $p = \{(-, \cdots, -)_m, p_{m+1}\}$ 是利用变量在逻辑“与”和逻辑“或”运算时满足交换律和结合律，将所有取值为“-”的变量放在一

起的结果。同样也可以调整 H 中的变量顺序使得与 $p = \{(-, \dots, -)_m, p_{m+1}\}$ 一致。

(3)当 $n = m + 1, n \leq k$ 时。此时逻辑项 p 可以写成 $p = \{(-, \dots, -)_{m+1}, p_{m+2}\} = \{(-, \dots, -)_m, -, p_{m+2}\}$ 。显然 p 可以展开为下面 2 个逻辑项 $p_1 = \{(-, \dots, -)_m, 1, p_{m+2}\}$ 和 $p_2 = \{(-, \dots, -)_m, 0, p_{m+2}\}$ ，相应地极性表达式也可以表示为 $H = \{H_m, q, H_{m+2}\}$ 。利用 3.1 节中的串行列表技术步骤 1, 步骤 2 可得, 当 $q = 1$ 时, 可以得到 $p'_1 = \{(-, \dots, -)_m, 0, p'_{m+2}\}$, $p'_2 = \{(-, \dots, -)_m, 1, p'_{m+2}\}$, 进而可以将 p'_1 和 p'_2 合并为一项 $p' = \{(-, \dots, -)_m, -, p'_{m+2}\} = \{(-, \dots, -)_{m+1}, p'_{m+2}\}$ 。同理, 当 $q = 0$ 时也可以得到 $p'' = \{(-, \dots, -)_m, -, p'_{m+2}\} = \{(-, \dots, -)_{m+1}, p'_{m+2}\}$ 。即定理 1 在 $n = m + 1, n \leq k$ 时也成立。证毕

定理 2 若构成 f 的乘积项为不相交乘积, 则用列表技术将 f 从 AND/OR 形式转换为 FPRM 形式时可以采用的步骤为: (1)忽略不相交乘积项中取值为“-”的变量, 只对取值为“0”和“1”的变量执行 3.1 节中的并行列表算法步骤 2~步骤 4 的操作; (2)去掉重复项后把剩余项中取值为“-”的变量变为“0”, 最后成对删去重复项, 剩余的项即为所求极性下的 RM 表达式的项。

下面用数学归纳法证明定理 2。考虑到 2 个乘积项的 AND/OR 形式向零极性 AND/XOR 形式转化过程可以推广到多个乘积项的 AND/OR 形式向任意极性的 AND/XOR 形式转化, 因此本文只给出 2 个不相交乘积项 p, q 的 AND/OR 形式向零极性下 AND/XOR 形式转化过程中上述定理正确性的证明。

证明 假设 x_1, x_2, \dots, x_k 为 k 个输入变量, 不相交乘积项 p, q 中分别有 n 和 m 个“-”, $0 < n < k, 0 < m < k$ 。下面用数学归纳法证明用定理 2 提出的方法得到 AND/XOR 表达式和从已发表的基于最小项列表法进行转换具有相同的结果。同时考虑到 $0 \oplus \dot{x} = \dot{x}, x \in \{x_1, x_2, \dots, x_k\}$, 因此在证明过程中省略了乘积项与所求极性表达式进行异或操作这个步骤。

(1)当 $n = m = 1$ 时 假设 p 中的 x_i, q 中的 x_j 取值为“-”, 则 p, q 可以表示为

$$\left. \begin{aligned} p &= \{p_{i-1}, -, p_{i+1}\} \\ q &= \{q_{j-1}, -, q_{j+1}\} \end{aligned} \right\} \quad (6)$$

式(6)中, $p_{i-1}, p_{i+1}, q_{i-1}$ 和 q_{i+1} 的含义同式(5)。根据串行列表法, 从表 3 可以得到 $f = p + q$ 对应的零极性 RM 表达式为 $f = \{p_{i-1}, 0, p_{i+1}\} \oplus \{q_{j-1}, 0, q_{j+1}\}$ 。其中逻辑项 $\{p_{i-1}, 0, p_{i+1}\}$ 和 $\{q_{j-1}, 0, q_{j+1}\}$ 中的变量之

表 3 $n=m=1$ 时 p, q 展开

p, q	展开“-”	执行步骤 2 到 4
$\{p_{i-1}, -, p_{i+1}\}$	$\{p_{i-1}, 1, p_{i+1}\}$	
	$\{p_{i-1}, 0, p_{i+1}\}$	$\{p_{i-1}, 1, p_{i+1}\}$
$\{q_{j-1}, -, q_{j+1}\}$	$\{q_{j-1}, 1, q_{j+1}\}$	
	$\{q_{j-1}, 0, q_{j+1}\}$	$\{q_{j-1}, 1, q_{j+1}\}$

间是逻辑“AND”关系。

如按照定理 2 提出的方法, 直接将逻辑项中的“-”用“0”来代替, 可以得到 $f' = \{p_{i-1}, 0, p_{i+1}\} \oplus \{q_{j-1}, 0, q_{j+1}\}$ 。显然 $f = f'$ 。即定理在 $n = m = 1$ 时成立。

(2)假设当 $n = w, m = v$ 时命题成立, 即

$$\begin{aligned} f &= p\{(-, \dots, -)_w, p_w\} + q\{(-, \dots, -)_v, q_v\} \\ &= p\{(0, \dots, 0)_w, p_w\} \oplus q\{(0, \dots, 0)_v, q_v\} \end{aligned} \quad (7)$$

为了描述方便, 在式(7)中, 利用逻辑“或”满足交换律, 将所有取值为“-”的变量放在一起。其中 $(-, \dots, -)_w$ 表示有 w 个“-”, 而 $(-, \dots, -)_v$ 表示有 v 个“-”, p_w, q_v 的含义同式(5)。

(3)当 $n = w + 1, m = v + 1$ 时, 部分展开 p 和 q , 得到

$$\begin{aligned} f &= p\{(-, \dots, -)_{w+1}, p_{w+1}\} + q\{(-, \dots, -)_{v+1}, q_{v+1}\} \\ &= p\{(-, \dots, -)_w, -, p_{w+1}\} + q\{(-, \dots, -)_v, -, q_{v+1}\} \\ &= p\{(-, \dots, -)_w, 0, p_{w+1}\} + p\{(-, \dots, -)_w, 1, p_{w+1}\} \\ &\quad + q\{(-, \dots, -)_v, 0, q_{v+1}\} + q\{(-, \dots, -)_v, 1, q_{v+1}\} \end{aligned} \quad (8)$$

结合式(7)和 3.1 节串行列表算法中的步骤 3 和步骤 4, 可以得到式(8)对应的 AND/XOR 的表达式为

$$\begin{aligned} f &= p\{(-, \dots, -)_w, 0, p_{w+1}\} + p\{(-, \dots, -)_w, 1, p_{w+1}\} \\ &\quad + q\{(-, \dots, -)_v, 0, q_{v+1}\} + q\{(-, \dots, -)_v, 1, q_{v+1}\} \\ &= p\{(0, \dots, 0)_w, 0, p_{w+1}\} \oplus p\{(0, \dots, 0)_w, 1, p_{w+1}\} \\ &\quad \oplus q\{(0, \dots, 0)_v, 0, q_{v+1}\} \oplus q\{(0, \dots, 0)_v, 1, q_{v+1}\} \\ &= p\{(0, \dots, 0)_w, -, p_{w+1}\} \oplus q\{(0, \dots, 0)_v, -, q_{v+1}\} \end{aligned} \quad (9)$$

再利用 $n = m = 1$ 时的结果, 可以将式(9)转化为

$$\begin{aligned} f &= p\{(0, \dots, 0)_w, -, p_{w+1}\} \oplus q\{(0, \dots, 0)_v, -, q_{v+1}\} \\ &= p\{(0, \dots, 0)_w, 0, p_{w+1}\} \oplus q\{(0, \dots, 0)_v, 0, q_{v+1}\} \\ &= p\{(0, \dots, 0)_{w+1}, p_{w+1}\} \oplus q\{(0, \dots, 0)_{v+1}, q_{v+1}\} \end{aligned} \quad (10)$$

证毕

从式(10)可得, 当 $n = w + 1, m = v + 1$ 时, 定理 2 也成立。综上可得定理 2 在 AND/OR 向零极性 AND/XOR 转化时成立。同理也可以证明上述的方法在 AND/OR 表达式向任意极性 AND/XOR 转化时也成立。

例 2 对函数 $f(x_0, x_1, x_2) = \bar{x}_1 \bar{x}_2 + x_0$ 分别用最
小项和不相交乘积项由 AND/OR 表达式到极性为
3 时 RM 表达式的转换。如表 4 和表 5 所示。得到
极性为 3 时 RM 表达式为: $f(x_0, x_1, x_2) = \oplus \sum(3, 4, 7)$ 。

表 4 基于最小项转换

乘积项	最小项	要求 极性	新最 小项	新项		
				x_0	x_1	x_2
-00	000	011	011	111	111	101
	100		111	110		
1--	101	011	110			
	110		101			
	111		100			

表 5 基于不相交乘积项转换

不相交乘积项	要求极性	新项	0 变 1 “-” 变 0
000	011	011	111
1--		1--	100

在表 5 中, 已经将逻辑函数 f 转换成不相交项
之“或”。即 $f(x_0, x_1, x_2) = \bar{x}_1 \bar{x}_2 + x_0 = \bar{x}_0 \bar{x}_1 \bar{x}_2 + x_0$ 。
由表 5 可得极性为 3 时 AND/XOR 表达式为: $f(x_0,$
 $x_1, x_2) = \oplus \sum(3, 4, 7)$ 。

显然用不相交乘积项和用最小项转换的结果是
相同的, 但是采用不相交乘积项的列表技术后, 无
论是待处理的乘积项数量, 还是每个乘积项中的变
量的数量都将大大减少。因此基于不相交项的列表
技术有更高的效率, 并且适合处理更大的电路。

3.3 基于不相交项的并行列表技术

鉴于 3.2 节的分析, 本文提出基于不相交乘积
项的并行列表法实现 AND/OR 向 FPRM 展开式的
转换。转换算法的主要步骤为:

步骤 1 把乘积项转换为不相交乘积项;

步骤 2 利用定理 1, 实现各不相交乘积项与
所要求的极性表达式进行位异或操作, 得到新的逻辑
项, 并用二进制表示;

步骤 3 设第 1 个新项有 s 个二进制位为 0,
则以这些位为无关项, 产生 $2^i - 1$ 个新项;

步骤 4 重复步骤 3, 直至操作完所有新项;

步骤 5 将新乘积项连同新项中的“-”变“0”;

步骤 6 成对删除相同的项, 剩下的乘积项即
为所求的固定极性下的 RM 逻辑函数项。

例 3 表 6 以一个 4 变量函数为例来说明乘积
项列表技术的具体步骤, 要求将函数展开为极性为
5 的 RM 表达式。函数的 Boole 表达式为

$$f(x_0, x_1, x_2, x_3) = x_0 \bar{x}_2 x_3 + \bar{x}_0 \bar{x}_1 x_3 + \bar{x}_0 x_2 \bar{x}_3$$

表 6 乘积项并行列表技术

乘积项 p $x_0 x_1 x_2 x_3$	极性表 达式 H	新乘积项 $p \oplus H$	新项	“-” 变 0
1-01	0101	1-00	1-11 1-10	1000 1011
			1-01	1010 1001
00-1	0101	01-0	01-1 11-0	0100 0101
			11-1	1100 1101
0-10		0-11	1-11	0011 1011

可以得到极性为 5 的 RM 表达式为

$$f(x_0, x_1, x_2) = \oplus \sum(3, 4, 5, 8, 9, 10, 12, 13)$$

4 实验结果及分析

本文算法用 C 语言描述, 在 VC++6.0 平台实
现。所用的计算机配置为 Windows XP 操作 3.29
GHz CPU 和 2.85 GB 内存。实验中, 本文与文献
[7]的方法进行了比较。文献[7]的方法是利用逻辑函
数输入冗余特性(input redundancy)实现对大函数
的从 AND/OR 展开式到 FPRM 展开式转换。因此
文献[7]方法的性能与待处理的逻辑函数的输入特
点或函数的结构有关。为了能较全面地衡量本文算
法的性能, 在实验中又另外选取了 11 个不同规模
的电路进行测试。

表 7 所示的是文献[7]方法与本文方法的比较结
果。表中列出了测试所用的电路, 电路的输入/输出
数/原始乘积项数($i/0/p$), 零极性下文献[7]方法与
本文方法各自对应的 FPRM 表达式中包含的 On-
set 个数, 以及算法运算的时间。表格中的时间单位
是 s。表中“<1 ms”表示算法显示的时间为 0。

表 7 中两种方法对应的 On-set 个数是一样的,
从一个侧面验证了本文方法的正确性。表 7 中最大
的测试电路的输入变量数为 199, 远远大于已发表
的列表法不超过 30 这个界限。表 8 给出了另外 11
个不同规模电路的测试结果。

综合表 7 和表 8 的结果, 不难发现本文算法的
特点: (1)由于避免以往列表法中必须以最小项为基
础实现 FPRM 的展开, 使得改进后的列表技术速度
更快, 同时更适合处理大逻辑函数; (2)本文方法运
算时间对待处理的逻辑函数的输入输出数量不敏
感, 但构成逻辑函数的乘积项的数量会影响本文方
法的运算速度。因此对于那些输入输出较大, 但是
乘积项数量不多的逻辑函数, 本文方法具有很高的
效率。

5 结论

传统列表技术在实现将 AND/OR 形式的逻辑
表达式向固定极性 RM 表达式转换过程中总是先将

表7 本文方法实现两种逻辑转换

电路	$i/0/p$	文献[7]On-set	文献[7]时间(s)	本文 On-set	本文时间(s)
apex6	135/99/657	11615	1090.62	11615	164.906
b9	41/21/138	706	0.94	706	0.015
c8	28/18/172	460	0.30	460	0.093
cht	47/36/120	178	0.04	178	0.015
count	35/16/184	131137	156.34	131137	0.109
example2	85/66/369	1076	0.81	1076	<1 ms
i6	138/67/236	341	0.10	341	0.015
i7	199/67/302	330	0.11	330	0.015
i8	133/81/3544	41874	158.26	41874	46.750
lal	26/19/102	745	0.48	745	<1 ms
misex2	25/19/29	1100	0.16	1100	<1 ms
pcler8	27/17/61	104	0.44	104	<1 ms
term1	34/10/257	9081	122.72	9081	8.421
unreg	36/16/48	132	0.03	132	0.015
x3	135/99/739	11615	1151.96	11615	27.078
x4	94/71/535	3174	2.10	3174	1.796

表8 不同规模电路测试结果

电路	$i/0/p$	On-set	时间(s)
alu2	10/6/261	294	0.125
apex7	49/37/517	5231	65.781
cc	21/20/52	59	<1 ms
clip	9/5/167	217	0.062
cu	14/11/22	230	<1 ms
examples	85/66/369	1076	0.828
il	25/13/27	1055	0.046
i5	133/66/369	6906	4.656
pcler	19/9/45	72	<1 ms
pm1	16/13/37	37	<1 ms
tcon	17/16/32	24	<1 ms

逻辑函数展开为最小项然后再进行其他操作。由于逻辑最小项的数量与输入变量的2的指数次方成正比相关，因此在处理大函数时，以往的基于最小项的列表方法效率很低甚至无法处理。针对以往列表法上述不足，本文结合并行列表技术，提出一种基于不相交乘积项的快速转换方法。该方法直接用不相交乘积项代替最小项，由于逻辑函数的不相交项的数量要远远少于其对应的最小项数量，因此提出的算法有效地避开了以往列表法在处理大函数时最小项数量激增导致的效率低下甚至无法工作的缺点。实验结果表明：本文提出的算法能处理的电路的规

模远远大于传统列表技术所能处理的电路。本文算法还具有对电路输入个数不敏感、对待处理电路的结构没有限制等特点。

参考文献

- [1] Almaini A E A, Thomson P, and Hanson D. Tabular techniques for Reed-Muller logic[J]. *International Journal of Electronics Theoretical and Experimental*, 1991, 70(1): 23-34.
- [2] Lozano C C, Falkowski B J, and Luba T. Fixed polarity linearly independent expansions for the representation of quaternary functions[J]. *Multiple-Valued Logic and Soft Computing*, 2012, 19(4): 307-324.
- [3] Castro J. Recent advances in optimization techniques for statistical tabular data protection[J]. *European Journal of Operational Research*, 2012, 216(2): 257-269.
- [4] Al Jassani B A, Urquhart N, and Almaini A E A. Manipulation and optimisation techniques for Boolean logic [J]. *IET Computers & Digital Techniques*, 2010, 4(3): 227-239.
- [5] 杨萌, 徐红英, Almaini A E. 针对混合极性的并行表格技术的遗传算法[J]. *计算机辅助设计与图形学学报*, 2011, 23(11): 1938-1943.
Yang Meng, Xu Hong-ying, and Almaini A E. Optimization of mixed polarity functions using genetic algorithm with parallel tabular technique[J]. *Journal of Computer-Aided Design & Computer Graphics*, 2011, 23(11): 1938-1943.
- [6] 汪迪生, 汪鹏君, 孙飞, 等. 包含无关项逻辑函数的固定极性

- 转换[J]. 电路与系统学报, 2013, (1): 117-121.
- Wang Di-sheng, Wang Peng-jun, Sun Fei, *et al.* Fixed-polarity conversions for logic functions include don't care terms[J]. *Journal of Circuits and Systems*, 2013, (1): 117-121.
- [7] Wang Lingli and Almani A E A. Fast conversion algorithm for very large Boolean functions[J]. *Electronics Letters*, 2000, 36(16): 1370-1371.
- [8] 卜登立, 江建慧. 基于混合多值离散粒子群优化的混合极性 Reed-Muller 最小化算法[J]. 电子与信息学报, 2013, 35(2): 361-367.
- Bu Deng-li and Jiang Jian-hui. Hybrid multi-valued discrete particle swarm optimization algorithm for mixed-polarity Reed-Muller minimization[J]. *Journal of Electronics & Information Technology*, 2013, 35(2): 361-367.
- [9] 王伦耀, 夏银水, 陈偕雄. 基于多数覆盖的二级 MPRM 函数逻辑优化[J]. 电子与信息学报, 2012, 34(4): 986-991.
- Wang Lun-yao, Xia Yin-shui, and Chen Xie-xiong. Two-level MPRM functions optimization based on majority cubes[J]. *Journal of Electronics & Information Technology*, 2012, 34(4): 986-991.
- [10] Wang Xiang, Lu Ying, Zhang Yi, *et al.* Probabilistic modeling during power estimation for mixed polarity Reed-Muller logic circuits[C]. IEEE International Conference on Green Computing and Communications, Beijing, China, 2013: 1414-1418.
- [11] Bernasconi A, Ciriani V, Drechsler R, *et al.* Logic minimization and testability of 2-SPP networks[J]. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2008, 27(7): 1190-1202.
- [12] Ciriani V. Synthesis of SPP three-level logic networks using affine spaces[J]. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2003, 22(10): 1310-1323.
- [13] Amaru L, Gaillardon P, and De Micheli G. MixSyn: an efficient logic synthesis methodology for mixed XOR-AND/OR dominated circuits[C]. 18th Asia and South Pacific Conference on Design Automation, Yokohama, Japan, 2013: 133-138.
- [14] Wang Lun-yao and Xia Yin-shui. Logic minimization based on dual logic[C]. 13th Conference on Computer-Aided Design and Computer Graphics, Hong Kong, China, 2013: 115-122.
- [15] Tan E C and Yang H. Fast tabular technique for fixed-polarity Reed-Muller logic with inherent parallel processes[J]. *International Journal of Electronics*, 1998, 85(4): 511-520.
- 王玉花: 女, 1987 年生, 硕士生, 研究方向为数字电路逻辑综合与优化.
- 王伦耀: 男, 1972 年生, 副教授, 研究方向为数字集成电路逻辑综合和优化、高信息密度集成电路设计.
- 夏银水: 男, 1963 年生, 博士, 研究生, 博士生导师, 研究方向为数字集成电路逻辑综合和优化以及 SoC 设计.