

基于不相交乘积项的逻辑探测和拆分算法

王伦耀^{1,2}, 夏银水¹, 陈偕雄², 叶锡恩¹

(1. 宁波大学信息科学与工程学院, 浙江宁波 315211; 2. 浙江大学信息与电子工程系, 浙江杭州 310027)

摘 要: 针对以往逻辑探测算法存在的不足, 提出了一种基于不相交乘积项的逻辑探测和拆分算法. 该算法通过将逻辑函数的乘积项转化为不相交乘积项, 并结合不相交乘积项之间海明距关系将不相交乘积项拆分成互不相交的二部分, 使之分别适合 RM 逻辑综合和布尔逻辑综合. 通过对拆分结果分析, 本文进一步提出了一种新的逻辑探测方法. 实验结果表明, 本文的判别结果能对电路综合过程中的逻辑选择提供一个良好的指导作用.

关键词: Reed-Muller 逻辑; 布尔逻辑; 逻辑探测; 海明距

中图分类号: TP391

文献标识码: A

文章编号: 0372-2112 (2012)10-2091-06

电子学报 URL: <http://www.ejournal.org.cn>

DOI: 10.3969/j.issn.0372-2112.2012.10.031

Logic Detection and Decomposition Algorithm Based on Disjointed Cubes

WANG Lun-yao^{1,2}, XIA Yin-shui¹, CHEN Xie-xiong², YE Xi-en¹

(1. School of Information Science and Engineering, Ningbo University, Ningbo, Zhejiang 315211, China;

2. Department of Information Science & Electronic Engineering, Zhejiang University, Hangzhou, Zhejiang 310027, China)

Abstract: With the deficiency of the published methods in the logic detection, a method of the logic detection and decomposition using the disjointed cubes and the Hamming distance is proposed. By employing the method, a cover of a logic function is decomposed into two disjointed parts. One is suitable for the Boolean logic implementation and the other is suitable for the Reed-Muller(RM) logic implementation. Furthermore, a novel approach for the logic detection is also presented in this paper. The experimental results show that the proposed method can give a quick judgment for a circuit to choose a proper logic during the logic synthesis for the most cases.

Key words: Reed-Muller logic; Boolean logic; Logic detection; Hamming distance

1 引言

数字逻辑电路既可以采用基于“与-或-非”运算的布尔(Boolean)逻辑, 也可以采用基于“与-异或”运算的Reed-Muller(RM)逻辑来实现. 但是对于同一电路而言, 采用不同的逻辑来表示, 其复杂程度差异很大. 如对于一个 n 变量的奇偶校验函数, 如用 RM 逻辑来实现, 仅为 n 个文字(literals); 而用传统的布尔逻辑来表示则需要 2^n 个文字. 同样对于那些适合布尔逻辑表示的逻辑电路如果用 RM 逻辑来综合也达不到好的优化结果. 因此, 在对任意给定的电路综合前, 应该先进行一些必要的分析和判断, 以确定采用何种逻辑来实现电路功能, 从而使电路的结构得到尽可能的优化. 这种预先的分析和判断就是本文所关注的逻辑探测. 事实上, 大多数电路本身就是布尔逻辑和 RM 逻辑的混合体. 因此, 多数

情况下采用单一的布尔或 RM 逻辑进行综合, 往往得不到好的优化结果, 所以理想的电路综合策略是采用布尔逻辑和 RM 逻辑相结合的双逻辑综合. 对于适合双逻辑综合的电路, 往往需要将给定的电路所对应的覆盖分成二部分, 使之分别适合用布尔逻辑综合和 RM 逻辑综合. 这样做的优点是: 可以直接借助现成的布尔逻辑综合优化工具, 如 espresso^[1] 和 RM 逻辑的面积或功耗优化技术^[2~5] 来实现整个电路的优化. 上述将电路分成二部分的过程就是本文要研究的逻辑拆分. 逻辑拆分是实现双逻辑综合的一个重要环节.

近来在逻辑探测方面已经开展了不少研究. 采用的方法包括逻辑函数拆分法^[6], 最小项探测法^[7], 最简项探测法^[8], 基于 BDD 的图形探测法^[9] 以及采用多数覆盖的逻辑探测方法^[10] 等. 上述方法中, 文献[6]的方法由于没有对原始乘积项进行预处理, 使得探测的结果受

原始乘积项构成的影响很大.文献[7]的方法由于利用最小项进行探测,而逻辑函数的最小项的数量与输入变量的数量成指数级关系,因此该方法在处理大函数时往往效率很低甚至无法处理.文献[8]的方法在运用时有一个前提,即逻辑函数已经转化成最简项的集合.但这个前提条件实际上是逻辑综合和优化的一个主要目标,确保这个前提条件是有难度的.文献[9]的方法只给出了变量固定展开顺序下的判别结果而并没有就变量的展开顺序与判别结果的关系进行说明.因此,面对同一个函数,不同的展开方式可能会得到不同的判断结果.此外,当输入变量的增加时,BDD会变得很庞大,这将使文献[9]的方法的效率面临挑战.

在判别标准上,文献[6~8]提出的判断标准与满足各自提出的判断模式的乘积项数有关;文献[9]提出的标准与“特征三角形”在 BDD 上覆盖的面积有关;文献[10]提出了基于覆盖率的判别标准.上述几种方法都仅给出一个统计数据或判断结果,基本上没有给出函数 f 中哪些部分适合进行 RM 综合,哪些部分适合进行布尔综合,这给以后的双逻辑综合带来困难.

在本文中,将在不相交乘积项的基础上提出逻辑探测算法.该算法在给出相应的探测结果的同时也给出逻辑拆分结果,即将原来的覆盖拆分成适合布尔和 RM 综合的二部分.由于本文的方法对初始的乘积项进行预处理,使得最终的判断结果并不依赖逻辑函数的初始乘积项;另外,由于整个逻辑函数被分成互不相交的二个部分,为以后开展的双逻辑综合优化提供方便.

2 定义

一个 n 变量逻辑函数 f 总可以写成式(1)的形式.

$$f = \bigcup_{i=0}^{k-1} p_i \quad (1)$$

式(1)中, p_i 为乘积项, k 为乘积项的个数,“ \cup ”表示乘积项之间是逻辑“或”关系.并记 f 的乘积项的集合为 S_f .若乘积项 $p_i \in S_f$,则 p_i 可以写成式(2)的形式.

$$p_i = x_0 x_1 \cdots x_j \cdots x_{n-1} \quad (2)$$

其中 $x_j \in \{0, 1, -\}$,分别表示 x_j 取反变量,原变量和该变量不出现.

定义 1 对于 2 个变量 $x_i, x_j \in \{0, 1, -\}$,它们相交运算记作 $x_i \cap x_j$.显然有 $0 \cap 0 = 0, 1 \cap 1 = 1, 0 \cap - = 0, 1 \cap - = 1, - \cap - = -, 1 \cap 0 = \emptyset$.

定义 2 对于 2 个乘积项 p_w 和 $p_v, (p_w, p_v) \in S_f, w \neq v$,则 p_w 和 p_v 的相交运算可以表示为:

$$p_w \cap p_v = \{(x_{w0} \cap x_{v0}), \dots, (x_{w(n-1)} \cap x_{v(n-1)})\} \quad (3)$$

从定义 2 可得,当乘积项 p_w 和 p_v 的第 i 位上的变量 x_{wi} 和 $x_{vi}, x_{wi}, x_{vi} \in \{0, 1\}, i \leq (n-1)$,满足 $x_{wi} \cap x_{vi} = \emptyset$ 时,则表示 p_w 和 p_v 不相交,记为 $p_w \cap p_v = \emptyset$.但若乘积项

p_w 和 p_v 满足 $p_w \cap p_v = \{(x_{w0} \cap x_{v0}), \dots, (x_{w(n-1)} \cap x_{v(n-1)})\} = p_v$,则表示 p_v 被 p_w 包含,记为 $p_v \subseteq p_w$.

定义 3 对于一个给定的如式(1)所示的 n 变量逻辑函数 f ,假设乘积项 $(p_w, p_v) \in S_f, w \neq v$,若 p_w 和 p_v 中对应的变量都出现,则 p_w 和 p_v 的海明距 $H(p_w, p_v)$ 可以表示为:

$$H(p_w, p_v) = \sum_{i=0}^{n-1} x_{wi} \wedge x_{vi} \quad (4)$$

式中“ \wedge ”为“异或”逻辑运算,且规定 $- \wedge - = 0$.

根据定义 3 可得,二个乘积项 p_w 和 p_v ,若它们包含的变量的个数不一样或对应的变量没有同时出现,则 p_w 和 p_v 之间不存在符合定义的海明距.

例如,假设 $p_w = \overline{x_0} x_1 \overline{x_3}, p_v = x_0 x_1 x_3$,则 $H(p_w, p_v) = (\overline{x_0} \wedge x_0) + (x_1 \wedge x_1) + (\overline{x_3} \wedge x_3) = 2$.而当 $p_w = \overline{x_0} x_1 \overline{x_3}, p_v = x_0 x_3$ 时,则计算它们间的海明距没有意义,因为变量 x_1 没有同时出现在 p_w 和 p_v 中.

3 算法实现

对于逻辑函数 $f = \bigcup_{i=0}^{k-1} p_i, p_w$ 和 p_v 为 f 的任意二个乘积项, $w \neq v$,若 p_w 和 p_v 互不相交,则有 $p_w \cap p_v = \emptyset$,由此可以得到式(5):

$$p_w + p_v = p_w \oplus p_v + p_w p_v = p_w \oplus p_v \quad (5)$$

因此对于由互不相交乘积项构成的逻辑函数 f 可以写成如下的形式:

$$f = \bigoplus_{i=0}^{k-1} p_i \quad (6)$$

式(6)中“ \oplus ”表示各乘积项之间是逻辑“异或”关系.

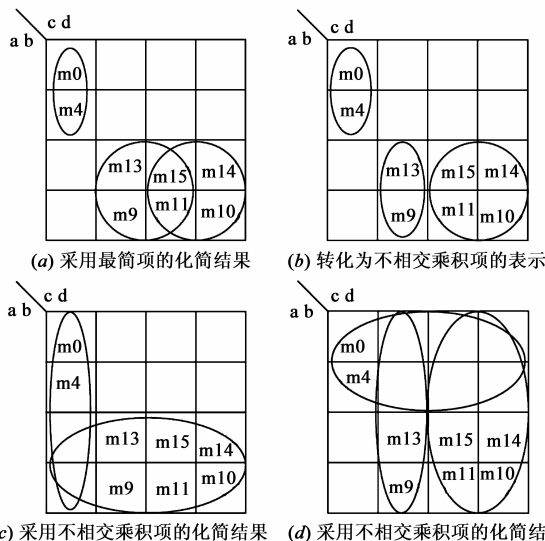


图1 乘积项构成对探测结果影响

互不相交乘积项既具有最小项互不相交的特点,又具有最简项一些特征,即一个不相交乘积项往往包含了多个最小项.这使得基于不相交乘积项的探测算

法既能克服最小项探测算法在处理大电路时效率低,又能弥补最简项探测算法的部分缺陷.如图 1(a)所示的逻辑函数 f ,基于不相交乘积项算法先将 f 分成 3 个不相交乘积项的组合,即 $\{\overline{acd}, \overline{acd}, ac\}$. f 最终可以简化成为图 1(c)或图 1(d)所示的形式.这就克服了文献[8]中用最简项无法判别的缺陷.

3.1 不相交乘积项生成算法

设二个乘积项 p_w 和 p_v ,它们对应的覆盖分别是 c_w 和 c_v ,并用符号“ \rightarrow ”表示二者的对应关系,即 $p_w \rightarrow c_w, p_v \rightarrow c_v$.令覆盖 c_d 等于 c_w 和 c_v 不相交锐积^[11],记作:

$$c_d = c_w \otimes c_v \quad (7)$$

其中(1)当 $p_w \cap p_v = \emptyset$ 时, $c_d = c_w$;

(2)当 $p_w \subseteq p_v$ 时, $c_d = \emptyset$.

当不属于(1),(2)情况时,若存在变量 $x_{wi} \subseteq p_w, x_{vi} \subseteq p_v$,且 $x_{wi} \otimes x_{vi} = \alpha_i, \alpha_i \in \{0,1\}$,则

$$c_d = c_w \otimes c_v = \bigcup_i f_i \quad (8)$$

其中 $f_i = (x_{w1} \cap x_{v1})(x_{w2} \cap x_{v2}) \cdots (x_{w(i-1)} \cap x_{v(i-1)}) \cdot \alpha_i \alpha_{i+1} \cdots \alpha_n$.

如假设 $p_w = (-0--), p_v = (1-01), p_w \rightarrow c_w, p_v \rightarrow c_v$,则 $c_d = c_w \otimes c_v = (00--, 101-, 1000)$,共由 3 个乘积项组成.这 3 个乘积项之间没有交集,与 p_v 也不相交.并且有 $c \cup c_v = c_w \cup c_v$.因此对于乘积项集合 S_f ,通过乘积项间逐个进行不相交锐积操作,可以将 S_f 转化为不相交乘积项集合,同时保持转化前后对应的逻辑覆盖不变.

逻辑函数的不相交乘积项生成算法的步骤如下:

(1)记 f 的 k 个乘积项的集合为 S_f, S_f 对应的覆盖集合为 $C, C = \{c_{p0}, c_{p1} \cdots c_{p(k-1)}\}; C_1$ 和 c_2 为二个覆盖,并令 $C_1 = \emptyset, c_2 = \emptyset$.

(2)依次从 C 中取出一个 $c_{pi}, i = 0, 1, \dots, (k-1)$.

令 $c_2 = c_{pi} \otimes (\bigcup_{j=i+1}^{k-1} c_{pj})$;

(3)令 $C_1 = C_1 \cup c_2$.重复执行步骤(2)和(3),直到 $i = k-1$.当 $i = k-1$ 时, $C_1 = C_1 \cup c_{p(k-1)}$;

(4)合并 C_1 中海明距为 1 的乘积项.

现以图 1(a)所示的逻辑函数 f 为例说明不相交乘积项的生成过程.由图 1(a)的卡诺图可得到函数 f 的覆盖为 $C = \{c_{p0}, c_{p1}, c_{p2}\} = \{0-00, 1--1, 1-1-\}$.由于 c_{p0} 与 c_{p1}, c_{p2} 都不相交,因此,执行上面步骤后得到 $c_2 = c_{p0} \otimes (c_{p1}, c_{p2}) = c_{p0}, C_1 = C_1 \cup c_2 = \{c_{p0}\} = \{0-00\}$,同时 C 更新为 $C = \{c_{p1}, c_{p2}\} = \{1--1, 1-1-\}$.再次执行上面步骤(2),得到 $c_2 = c_{p1} \otimes c_{p2}$.因为 c_{p1}, c_{p2} 相交,可以利用式(7)得到 $c_2 = c_{p1} \otimes c_{p2} = \{1-01\}, C_1 = C_1 \cup c_2 = \{c_{p0}, c_2\} = \{0-00, 1-01\}$,同时 C 更新为 $C = \{c_{p2}\} = \{1-1-\}$.考虑到此时 C 中只有一个元素 c_{p2} ,所以可

直接将 c_{p2} 添加到 C_1 ,最终得到 $C_1 = \{0-00, 1-01, 1-1-\}$,如图 2(b)所示.在本文中用 $gen_disjoint_p(C, C_1)$ 来表示不相交乘积项生成算法.

3.2 基于不相交乘积项的逻辑探测和拆分算法

当二个乘积项 p_w 和 p_v ,若 $H(p_w, p_v) = 2$,则有

$$p_w + p_v = p_w \oplus p_v = p'_w \oplus p'_v \quad (9)$$

式(9)中,生成的乘积项 p'_w 和 p'_v 所包含的字母数要小于乘积项 p_w 和 p_v .如假设 $p_w = \overline{ab}, p_v = \overline{ab}$,则 $p_w + p_v = \overline{ab} + \overline{ab} = \overline{ab} \oplus \overline{ab} = a \oplus b = p'_w \oplus p'_v$.显然采用 RM 逻辑的表达式更简单.

另外,当二个乘积项 p'_w 和 p_z ,若存在 $H(p'_w, p_z) = 1$,则有

$$p'_w + p_z = p'_w \oplus p_z = p_x \quad (10)$$

在式(10)中,不但乘积项的个数减少,新生成的乘积项中包含的字母数也减少了.

现假设逻辑函数 $f = \bigcup_{i=0}^{k-1} p_i$,且构成 f 的乘积项由 $gen_disjoint_p(C, C_1)$ 生成的不相交乘积项.则由式(6)

可得 $f = \bigcup_{i=0}^{k-1} p_i = \oplus \sum_{i=0}^{k-1} p_i$.假设这些 k 个不相交乘积项中,有 m 个乘积项之间的海明距满足式(9)和式(10),则式(6)可以转化成如下表达式:

$$f = \bigcup_{i=0}^{k-1} p_i = (\bigcup_{i=0}^{l-1} p_i) + (\sum_{j=0}^{m-1} p_j) = (\bigcup_{i=0}^{l-1} p_i) + (\oplus \sum_{j=0}^{h-1} p_j) \quad (11)$$

式(11)中,存在 $m + l = k, h \leq m$,且 $(\oplus \sum_{j=0}^{h-1} p_j)$ 包含的

字母数小于或等于 $(\sum_{j=0}^{m-1} p_j)$ 包含的字母数.实际上,式(11)将逻辑函数 f 分成了二部分,使之分别适合传统布尔逻辑实现和 RM 逻辑实现.因此,如果对 f 进行单一逻辑综合,则通过对式(11)拆分后结果进行分析有助于在综合过程中逻辑的选择,进而获得比较简单的综合结果.倘若对 f 进行双逻辑综合,则对式(11)拆分后形成的二部分逻辑覆盖分别用传统布尔逻辑和 RM 逻辑进行综合则可以获得较好的综合效果.在本文中,上述的逻辑探测和拆分过程由下面算法实现.

令 f 对应的不相交乘积项的覆盖为 $C_1, C_1 = \{c_{p0}, c_{p1} \cdots c_{p(k-1)}\}$.并令集合 $C_R = \emptyset, C_B = \emptyset$.算法步骤如下;

(1)将函数 f 的不相交乘积项集合 C_1 构成一个链表 L ,并清零各个乘积项的标志位;

(2)计算 C_1 对应的最小项的个数,记为 $minterms_C_1$;

(3)依次判别 L 中各个乘积项之间海明距,如果存在 $HD(p_i, p_j) = 2$,则将 p_i 简化为 R_i, p_j 简化为 R_j .将 R_i

和 R_j 存储在 L 最后, 并将 R_i 和 R_j 的标志位置为“1”. 检测 p_i, p_j 的标志位, 并将标志位为“0”的乘积项保存到 C_R 中. 在 L 中删除 p_i, p_j ;

(4) 如果 $HD(p_i, p_j) = 1$, 则将 p_i, p_j 合并为一项 B_i . 将 B_i 存储在 L 最后, 并将 B_i 的标志位置为“1”. 检测 p_i, p_j 的标志位, 将标志位为“0”的乘积项保存到 C_R 中. 在 L 中删除 p_i, p_j ;

(5) 循环执行(3), (4)直到没有新的乘积项的产生;

(6) 计算 C_R 对应的最小项的个数, 记为 $minterms_{C_R}$;

(7) 统计 L 中标志位为“1”的乘积项数, 记为 N_{PR} .

上述程序用 $Detc_decomp(C_1, C_R, C_B)$ 来表示. C_1 为不相交乘积项的集合. C_R 为适合 RM 逻辑综合的乘积项集合. C_B 为适合布尔逻辑综合的乘积项集合, 它对应算法运行结束后, C_1 中标志位为 0 的乘积项集合.

在上面步骤中, 虽然通过 $gen_disjoint_p(C, C_1)$ 生成的不相交乘积项已经将海明距为 1 的乘积项进行了合并, 但由于程序 $Detc_decomp(C_1, C_R, C_B)$ 会生成新的乘积项, 因此存在 $HD(p_i, p_j) = 1$ 的可能, 需要进行合并以减少乘积项的个数. 另外, 由于标志位为“1”的乘积项都是由原始的乘积项产生, 这些原始的乘积项在合并成新的乘积项的同时, 都从 L 中移入到 C_R 中, 所以, 标志位为“1”的乘积项对应的覆盖就是 C_R .

步骤(3)中的新乘积项 R_i, R_j 用下面方法生成:

(1) 确定乘积项导致 $HD(p_i, p_j) = 2$ 的二个变量, 设为 x_a 和 x_b ;

(2) 在 p_i 中保留 x_a 去掉 x_b ; 在 p_j 中去掉 x_a 保留 x_b . 修改后的 p_i, p_j 分别对应 R_i, R_j .

步骤(4)中新的乘积项 B_i 用下面方法生成:

(1) 确定导致 $HD(p_i, p_j) = 1$ 的变量, 设为 x_a ;

(2) 在 p_i 去掉 x_a , 修改后的 p_i 就对应 B_i .

4 实验及结果分析

在判别方法上, 本文通过比较适合 RM 综合的乘积项的覆盖率以及对应的乘积项数量的方法对逻辑函数 f 适合何种逻辑综合进行判断. 该方法包含二个部分:

(1) 计算 C_R 对应的最小项覆盖率, 用 C_v 来表示. 其中:

$$C_v = minterms_{C_R} / minterms_{C_1} \quad (12)$$

(2) 由于电路的复杂程度与对应的逻辑函数所包含的乘积项个数直接有关, 因此, 本文引入另外一个参数, 称为乘积项比例系数, 用 J 表示:

$$J = \lambda \frac{N_{PR}}{N_{P0}} \quad (13)$$

式(13)中, N_{PR} 表示程序 $Detc_decomp(C_1, C_R, C_B)$ 中标志

位为“1”的乘积项个数, N_{P0} 表示原始逻辑函数中乘积项的个数, λ 为一个比例系数, 本文用式(14)表示:

$$\lambda = \frac{1}{n} \sum_{i=0}^{n-1} \frac{(N_{P0})_i}{(N_{pes})_i} \quad (14)$$

式(14)中 $(N_{pes})_i$ 表示电路 i 经过二级布尔逻辑综合工具 espresso 优化后乘积项的个数, $(N_{P0})_i$ 表示电路 i 原始逻辑函数中乘积项的个数. 本文通过从 MCNC Benchmark 中随机挑选部分测试电路经过实验, 测得 λ 约为 1.62. 在实际应用中, 可以通过改变 λ 的大小来放宽或收缩相应的约束条件.

从式(12)(13)可以得到, 如果一个电路的 C_R 对应的 N_{PR} 比较小, 但同时 C_R 对应的覆盖率 C_v 又比较大的情况下, 那么该电路更加适合进行 RM 逻辑综合.

表 1 为本文算法的测试结果. 算法用 C 语言编程, 并在操作系统为 Windows Vista, CPU 为 2.2GHz, 内存为 2GB 的 PC 机上实现. 表 1 中的电路来自 MCNC Benchmark. 其中, 表 1 第 3 栏中数据为测试电路经过二级布尔逻辑综合工具 espresso 优化后对应的乘积项个数. 第 4 栏中数据为文献[12, 13]提供的最优极性下 RM 逻辑综合的结果. 第 5 栏 C_v 为根据式(12)计算所得的覆盖率.

为加快探测速度, 本文提出 2 个判断条件, 只要其中任何一个满足, 就认为该电路不适合采用单一的 RM 逻辑进行综合:

(1) $N_{PR} = 0$. 这个条件表示在被探测电路中没有找到一对可以构成 RM 逻辑的乘积项, 因此可以认定该电路不适合采用 RM 逻辑进行综合, 亦即可以判断该电路适合布尔逻辑综合, 所以没有必要再计算对应的 C_v 和 J 的值. 符合这种情况电路在表 1 中直接用“0”和“B”来表示.

(2) $N_{PR} > N_{P0}$. 这个条件表示采用单一的 RM 逻辑综合结果可能比原始的逻辑函数还要复杂. 考虑到此时 $N_{PR} > 0$, 因此可以直接判断该电路适合双逻辑综合, 所以也没有必要再计算具体的 C_v 和 J 的值, 相应地表 1 中用“-”和“D”来表示.

从表 1 可以看出, 有二个电路满足条件(1). 另外, 符合条件(2)的电路有 4 个, 其中电路 t481 被误判. t481 被误判与 t481 对应的函数结构有关. 它可以表示为 $f_{t481} = h \oplus g$ 形式, 其中 h 和 g 为 2 个子函数. 由于本文的探测算法是一个局部搜索算法, 无法探知 t481 可以表示成 $f_{t481} = h \oplus g$ 这种形式从而导致误判. 但从整个表 1 列出的 espresso 的综合结果与文献[12, 13]提供 RM 逻辑综合结果来看, 对于绝大多数电路, 本文的方法能给出一个比较准确的判断结果. 表 1 中最后一栏列出了本算法的运行时间. 其中“<1”表示算法显示的时间约为 0ms. 除电路 parity 外, 其余的电路运算速度都很快. 这

也从一个侧面反映了本算法的速度与要处理的乘积项的个数相关,而对函数的输入变量个数不敏感。

不同于文献[6~10]的方法,本文在下面二个方面进行了改进:

(1)对指定电路采用何种逻辑进行综合给出了一个综合的判断结果。

利用表 1 的数据可以得出三种探测结果.第一、对与那些 $C_v = 0$ 的电路,如果采用单纯的布尔逻辑进行综合,往往可以在电路面积上得到简化;第二、对于那些 $C_v = 1$,并且 $J < 1$,即 $N_{PR} \leq \frac{N_{PO}}{\lambda}$ 的电路,采用单纯的 RM 逻辑进行综合往往在电路面积上可以得到简化,如表 1 中 parity 和 xor5 等电路;第三、对于不符合上面 2 个条件,并且 $0 < C_v < 1$ 的电路,适合采用双逻辑综合.表 1 中多数电路属于这种情况.相比之下,文献[6~8]只给出了“YES”或“NO”的二种区分的判别结果.其中探测结果为“NO”的电路表示适合采用布尔逻辑综合;但是对于那些探测结果为“YES”的电路来说,这些文献没有进一步说明对应的电路到底是采用单一的 RM 逻辑还是采用布尔与 RM 逻辑混合的双逻辑进行综合。

表 1 几种逻辑电路的相应逻辑综合结果与探测结果的比较

电路名称	输入/输出	espresso 综合结果	RM 综合 结果 ^[12,13]	C_v	N_{PR}/N_{PO}	$J = \lambda \frac{N_{PR}}{N_{PO}}$	运行时间(ms)
5xp1	7/10	65	61	0.98	41/75	0.89	16
9sym	9/1	86	173	-	124/87	D	19
alu4	14/8	575	4334	-	1511/1028	D	684
apex4	9/19	436	512	0.05	18/438	0.07	< 1
b12	15/9	43	68	0.08	32/430	0.12	< 1
bw	5/28	22	22	0.92	9/87	0.17	< 1
clip	9/5	120	206	0.70	123/167	1.19	< 1
cm150a	21/1	17	82	-	26/17	D	< 1
con1	7/2	9	18	0.54	4/9	0.72	15
ex1010	10/10	284	1018	0.85	582/1024	0.92	62
duke2	22/29	284	267	0.95	31/87	0.58	< 1
life	9/1	84	184	0.27	80/140	0.93	15
inc	7/9	30	49	0.12	2/35	0.09	< 1
misex1	8/7	12	20	0	0/32	B	< 1
misex2	25/18	28	124	0	0/29	B	< 1
misex3	14/14	690	3610	0.53	878/1848	0.77	172
misex3c	14/14	197	1890	0.003	8/305	0.04	15
parity	16/1	32768	16	1.00	16/32768	0.0008	93441
rd73	7/3	127	63	0.98	69/141	0.79	8
rd84	8/4	255	164	0.98	169/256	1.06	15
sao2	10/4	58	100	0.66	40/58	1.12	< 1
t481	16/1	481	17	-	929/481	D	312
table5	17/15	158	2807	0.045	14/159	0.14	< 1
xor5	5/1	16	5	1.00	5/16	0.51	< 1
vg2	25/8	110	5290	0.91	74/110	1.09	16

另外,如果一个电路适合采用双逻辑进行综合,则需要对电路进行适当的拆分,以确定电路的哪些部分

用布尔逻辑综合,哪些部分用 RM 逻辑综合.但是文献[6~10]并没有给出拆分的结果或方法,从而给电路的双逻辑综合带来障碍.针对这个不足,本文给出了二个改进.

(2)给出了适合双逻辑综合的逻辑拆分结果。

本文提出的探测方法在给出探测结果的同时也将相应的逻辑函数拆分成二部分,即 C_R 和 C_B ,由于采用不相交乘积项,因此满足 $C_R \cap C_B = \emptyset$, $C_R \cup C_B = C_1$. 其中 C_R 对应的乘积项的集合适合用 RM 逻辑进行综合,而 C_B 对应的乘积项的集合适合用布尔逻辑进行综合, C_1 对应的乘积项的集合为函数 f . 因此,最终的逻辑函数可以表示为 $f = f_B + f_R$,其中 $f \rightarrow C_1$, $f_B \rightarrow C_B$, $f_R \rightarrow C_R$. 由于逻辑函数被拆分成互不相交的二部分,因此可以采用现成的布尔逻辑或 RM 逻辑综合技术对相应的乘积项的集合分别进行逻辑综合与优化,进而实现给整个逻辑函数优化。

5 结论

本文利用不相交乘积项之间的逻辑“或”和逻辑“异或”可以互换的特性,提出了基于不相交乘积项的逻辑拆分和判断算法.不同于以往的探测方法,本算法给出了更加细分的判断结果,即将电路分为 3 类,它们分别适合布尔逻辑综合, RM 逻辑综合和双逻辑综合.同时,本算法将原来的逻辑函数对应的乘积项集合拆分成互不相交的二部分,其中一部分适合进行 RM 逻辑综合,另一部分适合进行布尔综合.基于这种拆分,就可以针对性地运用一些现成的综合技术来分别实现不同部分的面积和功耗优化,从而达到深入优化整个电路的目的.由于采用不相交乘积项,因此提出的探测算法兼具基于最小项和最简项逻辑探测算法的特点,同时又能较好解决基于最小项的算法在探测大电路失效的问题.另外,本文的算法只是涉及 ONSET 探测,而尚未对 OFFSET 探测进行探讨,但在有些情况下,用 OFFSET 进行综合结果更加简单,花费时间也更少.这些都是我们下一步的研究工作。

参考文献

- [1] P McGeer, J Sanghavi, R Brayton, et al. Espresso-signature: A new exact minimizer for logic functions [J]. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 1993, 1(4): 432 - 440.
- [2] L Wang, Y Xia, X Chen, et al. Reed-Muller function optimization techniques with onset table [J]. Journal of Zhejiang University Science-C, 2011, 12(4): 288 - 296.
- [3] Y Xia, X Wu, A E A Almaini. Power minimization of FPRM functions based on polarity conversion [J]. Journal of Computer

- Science and Technology, 2003, 18(3): 325 – 331.
- [4] 汪鹏君, 李辉, 吴文晋等. 量子遗传算法在多输出 Reed-Muller 逻辑电路最佳极性搜索中的应用[J]. 电子学报, 2010, 38(5): 1058 – 1063.
Wang Peng-jun, Li Hui, Wu Wen-jin, et. al. Application of quantum genetic algorithm in searching for best polarity of multi-output Reed-Muller logic circuits [J]. Acta Electronica Sinica, 2010, 38(5): 1058 – 1063. (in Chinese)
- [5] 王伶俐, 陈偕雄, 吴训威. 模为合数时多值模代数的模减与模除运算[J]. 电子学报, 1998, 26(5): 17 – 20.
Wang Lin-li, Chen Xie-xiong, Wu Xun-wei. Modulo subtraction and modulo division in composite multivalued modulo-algebra [J]. Acta Electronica Sinica, 1998, 26(5): 17 – 20. (in Chinese)
- [6] Y Xia, F Sun, K Mao. A detection method for logic functions suitable for dual-logic synthesis [J]. Progress in Natural Science, 2009, 19: 1311 – 1315.
- [7] 夏银水, 毛科益, 叶锡恩. 逻辑函数适合双逻辑实现的探测算法[J]. 计算机辅助设计与图形学报, 2007, 19(12): 1522 – 1527.
Xia Yin-shui, Mao Ke-yi, Ye Xi-en. Detection algorithm for logic functions to benefit from dual logic implementation [J]. Journal of Computer-Aided Design & Computer Graphics, 2007, 19(12): 1522 – 1527. (in Chinese)
- [8] 叶锡恩, 毛科益, 夏银水. 基于乘积项的双逻辑实现探测算法[J]. 电子学报, 2009, 37(5): 961 – 965.
Ye Xi-en, Mao Ke-yi, Xia Yin-shui. Algorithms for detecting dual logic based on product term [J]. Acta Electronica Sinica, 2009, 37(5): 961 – 965. (in Chinese)
- [9] Sun F, Xia Y. BDD based detection algorithm or XOR-type logic [A]. Proceedings of the 11th IEEE International Conference on Communication Technology Proceedings [C]. Piscataway, IEEE Computer Society, 2008. 351 – 354.
- [10] L Wang, Y Xia, X Chen. Logic detection algorithm for dual logic implementations based on majority cubes [A]. Proceedings of the International Conference on Computer Application and System Modeling [C]. Piscataway, IEEE Computer Society, 2010. V14: 503 – 507.
- [11] 边计年, 薛宏熙, 苏明等. 数字系统设计自动化[M]. 第二版, 北京: 清华大学出版社, 2005.
Bian Ji-nian, Xue Hong-xi, Su Ming, et. al. Design Automation for Digital Systems [M]. Edition II, Beijing: Tsinghua University Press, 2005. (in Chinese)
- [12] Wang L, Almaini A E A. Exact minimization of large multiple output FPRM functions [J]. IEE Proceedings of Computers and Digital Techniques, 2002, 149(5): 203 – 212.
- [13] Aborhey S. Reed-Muller tree-based minimization of fixed polarity Reed-Muller expansions [J]. IEE Proceedings of Computers and Digital Techniques, 2001, 148(2): 63 – 70.

作者简介



王伦耀 男, 1972 年出生, 浙江宁波人, 副教授. 主要从事数字电路逻辑综合与优化以及低功耗数字逻辑电路设计方面的研究.

E-mail: wanglun Yao@nbu.edu.cn



夏银水 男, 1963 年出生, 浙江余姚人, 教授, 博士生导师. 主要从事集成电路设计自动化, Soc 和 IP 核设计方面的研究.

E-mail: xiayinshui@nubu.edu.cn