

# 组合逻辑自动综合的超前试探法

邓德祥 李玉林 钟诚东

(武汉测绘科技大学光电工程学院,武汉市珞喻路39号,430070)

**摘要** 自动逻辑综合是复杂组合电路设计中必须面对的问题之一。本文采用超前试探策略,在改进的 Q-M 法基础上,提出了一个新的组合自动综合的新算法,该算法提高了运算速度,有较强的实用价值。

**关键词** 组合逻辑;自动综合;超前试探

**分类号** TP301.6

在组合逻辑综合过程中,随着变量数的增加,卡诺图和代数法将无法胜任逻辑化简这一关键性任务,因而必须采用计算机进行辅助设计。目前已有不少组合逻辑化简算法,其中改进的 Q-M 法是一种有代表意义的较好算法。该算法在构成试探性立方时,其试探策略为:最大限度构成可能的最大立方,以保证尽快生成最大的质蕴涵项(Prime Implicant),覆盖尽可能多的最小项。但在具体应用时,由于计算工作量大,尤其是在输入变量数超过10的情况下,计算时间长(例如在586/133微机上,输入变量数为16,耗时32min)为此,本文介绍一种新的超前试探法,它的基本策略是:最大限度构成可能的最大立方,并超前假定它为蕴涵项(Implicant),然后直接进行质蕴涵项测试。该算法提高了组合逻辑综合的运算速度,具有较强的实用意义。

## 1 组合逻辑函数的立方表示法

布尔代数系统  $(B^2, +, *, \sim, 0, 1)$  中布尔格  $B^2$  可以用几何立方结构图表示  $n$  个变量的布尔格  $B^n$  构成了一个  $n$  维立方体。组合逻辑函数的立方表示法就是用所有函数值为1的顶点(维立方)所构成的布尔子格(简称立方)来表示该函数。另外,由于布尔格特定的格结构,因而只要我们确定了该格的最大元  $\max(c)$  和最小元  $\min(c)$  也就确定了该格的立方体  $c$ , 从而立方  $c$  可记为:

$$c = \{\max(c), \min(c)\}$$

**定义1** 设  $V$  和  $U$  分别为  $B^n$  的顶点,若  $V^* U = V$ , 则称  $U$  包含  $V$  记为  $U \supset V$ 。

**定理1** 对  $B^n$  中任意立方  $c$  有  $\max(c) \supset \min$

( $c$ )。

**定理2** 设  $V$  是  $B^n$  中的顶点,  $c$  为  $B^n$  中的立方。若  $V^* \min(c) = \min(c)$ ,  $V^* \max(c) = V$ , 则  $V$  被包含在立方  $c$  中,即  $V \subseteq c$ 。

**定理3**  $c_1, c_2$  分别为  $B^n$  中的立方,  $c \subseteq c_2$ , 当且仅当

$$\min(c_1) \rightarrow \min(c_2) \text{ 且 } \max(c_2) \rightarrow \max(c_1)$$

## 2 组合逻辑综合的超前试探算法

在构成试探性立方时,尽量先构成可能的最大立方,对于选定的一个极小最小项  $i$ , 极大最小项  $j$ , 进行试组立方,并超前假定它为蕴涵项,直接进行质蕴涵项测试。其具体步骤如下:

**A1** 判别逻辑函数  $f$  的维数  $n$ , 并寻找最小项,若最小项数  $m \geq 2 - 1$ , 则化简其补函数,随后按递增顺序构成最小项表 MIN TABLE。

**A2** 选择  $(i, j)$ , 其中  $i = \min\{\text{MIN TABLE}\}$ ,  $j = \max\{\text{MIN TABLE}\}$ 。

**A3** 如果  $i^* j = i$ , ( $j \supset i$ ) 则直接进行质蕴涵项测试。**A4**, 否则选择下一个数值最大的项为  $j$ , 重复 **A2**。

**A4** 质蕴涵项测试, 如果  $(i, j) \subseteq \text{PITABLE}$  (质蕴涵项表), 则选择下一个数值最大的项为  $j$ , 重复 **A2**, 否则 **A5**。

**A5** 生成立方  $(i, j)$  的所有顶点  $V$ , 如果  $\forall V \Rightarrow V \subseteq \text{MIN TABLE}$ , 则  $(i, j) \subseteq \text{PITABLE}$ 。如果  $i$  已与 MIN TABLE 中所有比  $i$  大的项都配对比较过, 则 **A6**, 否则选择下一个数值最大的项为  $j$ , 重复 **A2**。

**A6** 由于所有包含  $i$  项的质蕴涵项都在

PITABLE中,因而有: 如果包含  $i$  项的质蕴涵项数仅有一个,则该质蕴涵项进入基本质蕴涵项表 EPITABLE

- A7 选择下一个  $i$ ,重复 A2至 A6
- A8 求所有未被 EPITABLE所包含的最小项,并选择能包含这些最小项的最佳 PI集 这个 PI集加上 EPITABLE就代表该函数的最简式

### 3 算法分析

空间要求: 对逻辑函数  $f$ ,若其维数为  $n$ ,则 MINTABLE要求  $2^{n-1}$  个无符号整型 (Unsigned int), PITABLE 要求  $2^n$  个无符号整型, EPITABLE要求  $2^n$  个无符号整型。

时间要求: 设函数  $f$  的最小项和无关项数为  $m$  当  $m$  比较小时,因  $c_m^2$  较小,这里不予讨论 从 而不失一般性可设  $2^r/3 < m < 2^r/2$

- A3 记  $c_m^2$  中有 NCT个  $j \rightarrow i$ , NCF个  $j \rightarrow i$ ,即

按位与和比较运算次数  $N3= c_m^2= NCF+ NCT$ , 其淘汰率满足 (1)式:

$$Z_3= NCT/c_m^2 \geq (m-n)/c_m^2 \approx 2/(m-1) \tag{1}$$

A4 记 NCT个  $(i,j)$ 中有 NIT个满足  $(i,j) \subseteq$  PITABLE, NIF个不满足,其淘汰率为:

$$Z_4= NIF/NCT \tag{2}$$

A5 记按位与和比较运算次数为  $N5$ ,有  $N5= (2^r-m)NIF= c_m^2Z_3Z_4(2^r-m)$

由以上介绍不难 得,本文算法的总运算次数为:

$$N_T= c_m^2(1+ Z_3Z_4(2^r-m)) \tag{3}$$

改进的 Q-M 法的总运算次数为:

$$N_T^n= c_m^2(1+ Z_3(2^r-m)) \tag{4}$$

运算次数减少倍数:

$$K= N_T^n/N_T= \frac{(1+ Z_3(2^r-m))}{(1+ Z_3Z_4(2^r-m))} \tag{5}$$

现令  $n= 16,m= 32\,768$ ,则有表 结果

表 1 运算次数减少倍数  $K$

Tab. 1 Reducing Factor  $K$  for Operation Quantity

$1/m_3$	$1/m_4$									
	32 768	16 384	8 192	4 096	2 048	1 024	512	256	128	64
16 384	3.00	3.00	3.00	3.00	3.00	2.99	2.98	2.97	2.95	2.90
8 192	5.00	5.00	5.00	5.00	4.99	4.99	4.99	4.92	4.84	4.70
4 096	9.00	8.99	8.99	8.99	8.99	8.77	8.59	8.37	8.12	8.00
2 048	17.00	17.0	16.9	16.9	16.8	16.7	16.4	16.0	15.1	13.6

### 4 应用示例

认证译码器用于需要严格限制访问的系统中,只有在输入信号和内部状态的 步 序是正确的时刻,才允许对该系统进行访问。被解析的认证译码器是一个 40 管脚的专用芯片,其结构可以用两

极主从关系的状态机来描述。主状态机又叫主定序器,可以用它来检验从状态机的状态 从状态机 (从定序器)可测试输入和内部状态是否到达预定的状态。如图 1所示 (图中的符号 ERR代表不正确命令),主定序器是有 8个状态的状态机,当从定序器到达某一预定状态时,它就步进到另一状态。从定序器使用 7个译码法则来决定主定序器步进的状态。

图 2为认证译码器的方框图。从定序器监视输入  $D_0 \sim D_{15}$ 和内部状态  $R_0 \sim R_7$ ,同时也监视状态位  $S_0 \sim S_7$ 和 HOLD 在上电时,译码器和状态寄存器自动复位,从而使主定序器处于状态 A 中, HOLD有效,在时钟上升沿,核对第一译码法则,如符合,则 HOLD无效,并步进到状态 B,如不符合,则停留在状态 A 当主定序器处于状态 B,并且 HOLD重新有效时,译码器检查第二法则,当条件为真时, HOLD再次无效,主定序器步进到状态 C,否则,返回状态 A 依此类推,当主定序器步进到状态 H时,认证成功,允许访问 (CODE有

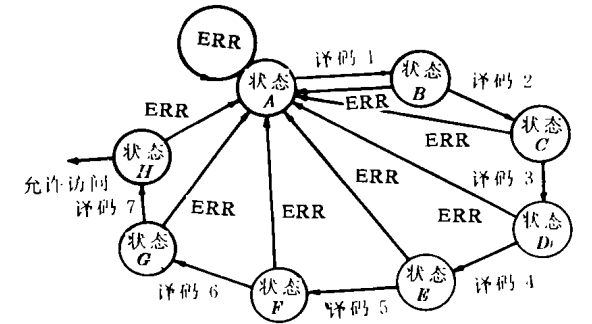


图 1 主定序器状态转换图

Fig. 1 State Transition Diagram of Main Sequencer

效)。

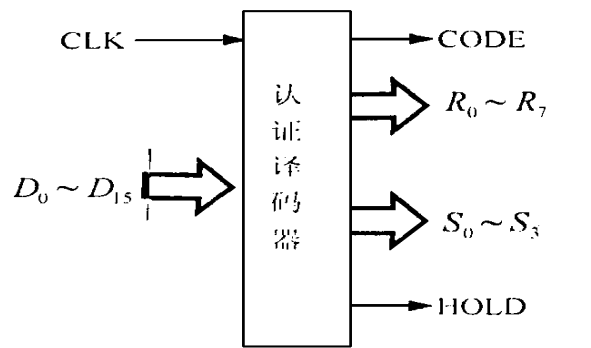


图 2 方框图

Fig. 2 Block Diagram

为解析 7 个译码法则,在不同状态时,以  $D_0\sim D_{15}$  和  $R_0\sim R_7$  为输入,以  $HOLD$  为输出,采用穷举法,获取 7 个真值表,分别用改进的 Q-M 法和本文方法进行计算,得 7 个逻辑表达式如下:

$$HOLD_1 = \overline{R_0} \overline{R_1} \overline{R_2} \overline{R_3} \overline{R_4} \overline{R_5} \overline{R_6} \overline{R_7}$$
$$HOLD_2 = \overline{D_0} \overline{D_1} \overline{D_2} \overline{D_3} + \overline{D_{12}} \overline{D_{13}} \overline{D_{14}} \overline{D_{15}}$$
$$HOLD_3 = \overline{R_0} \overline{R_1} \overline{D_0} \overline{D_{14}} \overline{R_6} \overline{R_7} \overline{D_{14}} \overline{D_{15}} + \overline{D_6} \overline{D_7}$$
$$HOLD_4 = \overline{D_8} \overline{D_9} \overline{D_{10}} \overline{D_{11}} \overline{D_{12}} \overline{D_{13}} \overline{D_{14}} \overline{D_{15}}$$
$$HOLD_5 = \overline{D_0} \overline{D_3} \overline{D_5} \overline{D_7} + \overline{D_2} \overline{D_{13}} \overline{D_{14}} \overline{D_{15}} + \overline{R_0} \overline{R_1} \overline{R_2} + \overline{R_5} \overline{R_6} \overline{R_7}$$
$$HOLD_6 = \overline{D_0} \overline{D_1} \overline{D_2} \overline{D_3} \overline{D_4} \overline{D_5} \overline{R_6} \overline{R_7} + \overline{R_0} \overline{R_1} \overline{R_2} \overline{R_3} \overline{R_4} \overline{R_5} \overline{R_6} \overline{R_7}$$
$$HOLD_7 = \overline{D_0} \overline{D_1} \overline{R_2} \overline{D_{12}} \overline{D_{13}} \overline{R_4} \overline{D_3} \overline{D_4} \overline{R_7}$$

解析出的表达式经实验验证,完全正确。表 2 列出的是在 586/13 微机上求解 7 个真值表所用的时间。从中可以看出,本文方法比改进的 Q-M 法所用时间均有所减少。其中在求解译码 5 时,所用时间减少近 5/倍。

表 2 运算时间比较表 (/s)

Tab. 2 Compare Table of Computing Time (/s)

	译码 1	译码 2	译码 3	译码 4	译码 5	译码 6	译码 7
改进的 Q-M 法	668	1 689	3 713	778	4 716	709	3 443
本文方法	519	752	907	677	834	503	1 578

参 考 文 献

1 杨士元.数字系统的故障诊断与可靠性设计.北京:清华大学出版社,1989.

2 孙涵芳,徐爱卿.可编程逻辑器件 PAL和 GAL.北京:北京航空航天大学出版社,1992.

3 现代工程数学手册,第一卷.武汉:华中工学院出版社,1985.

4 现代工程数学手册,第三卷.武汉:华中工学院出版社,1988.

5 Bossen D C, Hong S J Cause-Effect Analysis for Multiple Fault Detection in Combination Networks. IEEE Trans. On Computers, 1971, c- 20( 11): 1 252~ 1 257

6 Hayes J P. An Introduction to Switch-Level Modeling. IEEE Design & Test of Computers, 1987, 4(4): 18 ~ 25

7 Wolf W H How to Build a Hardware Description and Measurement System on an Object-Oriented Programming Language. IEEE Trans. On Computer-Aided Design, 1989, 8( 3): 288~ 301

8 Abadir M S, Ferguson J, Kirkland T T. Logic Design Verification via Test Generation. IEEE Trans. On Computer-Aided Design, 1988, 7(1): 138~ 148

An Advance Probe Method for Automatic Synthesize of Combinational Logic Circuits

Deng Dexiang Li Yulin Zhong Chengdong

(School of Photoelectric Engineering, W TU SM, 39 Luoyu Road, Wuhan, China, 430070)

**Abstract** Automatic synthesize is an important work in design of complicated combinational logic circuits. Based on improving Q-M method, a new method is proposed in this paper by using strategy of advance probe. The efficiency of computation is improved.

**Key words** combinational logic circuits; automatic synthesize; advance probe