

# 基于递归学习的组合电路等价性检验方法研究

曾 琼<sup>1,2</sup>

(1. 成都信息工程学院计算机基础教学部, 四川 成都 610225; 2. 中国科学院成都计算机应用研究所, 四川 成都 610041)

**摘要:** 等价性检验方法通常包括功能性和结构性的验证方法。分析了等价性检验的一般方法; 并讨论了基于递归学习的组合电路等价性检验方法; 算法利用直接蕴含和间接蕴含的方法, 解决了布尔可满足问题。实验结果表明了该方法的有效性和可行性。

**关 键 词:** 递归学习; 等价性检验; 组合电路

**中图分类号:** TP391.6

**文献标识码:** A

## 1 引言

随着集成电路的广泛应用, 对功能正确性及速度、功耗、可靠性等都有严格要求。其中, 功能正确性是最基本的要求。传统的模拟、测试和仿真等技术已经不能满足大型系统的设计验证。使用形式化验证方法有可能保证设计在各种可能的输入组合下的正确性。等价性检验是目前在工业实践中最广泛使用的形式化方法, 而且已被应用于验证大型复杂的设计<sup>[1]</sup>。等价性检验的主要目的是在一个设计经过变换之后, 穷尽地检验变化前后的功能的一致性, 即证明设计的变换没有产生功能的变化。它的基本原理是建立被比较的两个模型之间的关系。通常等价性检验程序能自动比较两个设计, 而不需要用户的输入。它的优点是使用简单, 且很容易集成到设计流程中。

具体的数字系统设计往往包含组合电路与时序电路两部分, 通常时序电路的等价性检验使用有限状态机遍历的方法来实现<sup>[2]</sup>。组合电路的等价性检验是指: 给定两个布尔网表, 检验它们的相应输出是否对所有可能的输入都相同。尽管验证两个组合电路的等价性在逻辑设计的 CAD 中非常重要, 然而这已被证明是一个 NP 完全问题。这表明要寻求一种完全地解决等价性检验的一般方法是很困难的。通常等价性检验方法的性能随着电路规模的增加而指数式地下降, 因此大多数实用方法都是利用分而治之的思想, 即通过某种策略将电路细分, 然后增量地验证两个电路的等价性, 从而能够处理较大规模的电路。

基于递归学习的组合电路等价性检验方法, 它利用直接蕴含和间接蕴含来解决布尔可满足问题, 即积自动机固定为 0 是否可测。直接蕴含是指一个门的扇出信号可以直接从前面或后面的门的扇出得到。间接蕴含是指其扇出信号有多种可能性, 需要同时考虑其前后多个门的逻辑信号才能确定。递归学习的方法可以认为是解决布尔可满足问题的通用的方法, 因此为逻辑验证提供了可选择的方法。我们对 ISCAS' 85 的部分电路进行了一些实验, 比较了递归学习方法和 BDD 方法的效率, 实验结果表明, 利用递归学习的方法对组合电路进行等价性检验能够有效地提高算法的效率。

## 2 等价性检验方法介绍

传统的组合电路功能等价性验证是通过构造两个电路的规范表示形式, 如真值表或二叉判定图 BDDs<sup>[1]</sup>, 当且仅当它们的规范形式同构时, 这两个电路功能等价。然而由于 BDD 的大小依赖于原始输入变量排序, 当电路规模较大时, 基于 BDD 的验证方法有可能引起内存爆炸。为了克服这种局限性, 一般采用基于结构相似性的增量验证方法。由于从设计流程的上一步到下一步, 两个设计版本之间往往保留了大量的结构相似性(即内部等价结点), 因此可利用这些相似性将整个电路的验证简化为一些小的验证任务, 增量地完成。首先使用名称信息、拓扑分析或随机模拟等方法产生候选的等价的结点, 然后按扇入优先的拓扑顺序, 逐步验证各候选结点的等价性或不等价性。常用于推导这些内部等价性的方法有 BDD, ATPG (Automatic Test Pattern Generation), SAT (Boolean Satisfiability)、可置换性及递归学习, 以及多种算法相结合的方法。总之, 增量验证方法的主要步骤可以分为 3 个

阶段: (1) 匹配候选对; (2) 对 miter 逐步剪枝, 简化 miter 的大小; (3) 检验每个原始输出对是否等价。

为了验证两个时序电路的等价性, 通常需要把它们当成有限状态机, 并构造二者的积自动机, 如图 1 所示。

Brand 将这种计算模型称为 miter<sup>[3]</sup>。它是通过把两个状态机相应的每一对原始输入联接到一起, 同时把相应的每一对原始输出联接到一个异或门, 而这些异或门就构成了积自动机的输出。如果对于每一个输入序列, 积自动机的每个原始输出恒为 0, 那么这两个时序电路就是等价的。换句话说, 就是对于任何输入向量和可达状态, 积自动机的原始输出响应总是为 0。通常, 证明状态机等价性的第一步是从初始状态开始, 计算所有可达状态。这就是典型的基于有限状态机遍历算法<sup>[4]</sup>。尽管最近十多年里, 由于 BDD 方法的研究进展使得基于有限状态机遍历算法有了很大进步, 但面对实际的大型设计仍然可能会因构造 BDD 的表示导致内存爆炸。

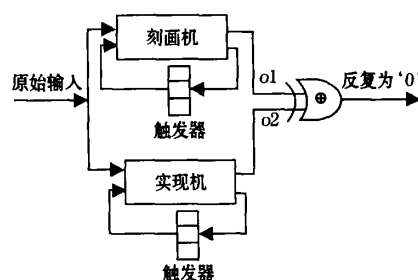


图1 积自动机(miter)

### 3 递归学习算法

递归学习方法是组合电路等价性检验方法中常用的方法, 它利用直接蕴含和间接蕴含解决布尔可满足问题。直接蕴含根据逻辑门的功能直接导出逻辑值, 如 FAN 算法中的蕴含<sup>[4]</sup>。间接蕴含则是通过对某个门赋一组逻辑值来找出所有一致的逻辑结果, 并将这些一致的逻辑值作为蕴含结果。如在 SOCRATE<sup>[5]</sup> 算法中就采用这种技术来辨识给定赋值情况下所有能唯一确定的信号值。当两电路的输入是相同一组信号线时, 如果一个电路的一些信号线赋值能导出另一电路中某些信号线赋值, 则意味着两电路中的信号线的值是相关的。功能相同的电路内部节点是这些相关信号线中的一部分, 可以将这些内部节点看作两电路的相似节点。找出电路的相似性后, 将电路在逻辑相等的内部节点处化分开, 验证问题分解为一个个子电路的验证问题。如何利用各种蕴含找出具有相同逻辑功能的内部节点是递归学习算法的关键, 也是提高验证效率的主要原因。下面介绍几个递归学习中用到的定义。

**定义 1** 给定门  $g$ , 至少有一个输入或输出已赋值, 这些赋值是相对于门  $g$  逻辑一致的。且对于几个门  $g$  的未赋值输入或输出, 存在不同的赋值组合, 在门  $g$  出现冲突, 称这样的门为未确认门。反之, 为确认门。

在图 2 中, (a) 和 (b) 的与门  $g$  的输出赋值都为 0, 一个输入已赋值, 另一个输入未赋值。图 2(a) 中与门  $g$  的一个输入赋值为 1, 对另一输入  $a$ , 有两种赋值可能, 当  $a$  为 1 时, 门  $g$  出现矛盾。因此, 门  $g$  是未确认门。图 2(b) 中与门  $g$  的一个输入赋值为 0, 对另一输入  $a$ , 无论赋为何值, 门  $g$  不可能出现矛盾, 故为确认门。

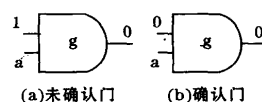


图2 逻辑门的确认

**定义 2** 一个未确认门  $g, f_1, f_2 \dots f_n$  是门  $g$  的一些未赋值的输入或输出,  $V_1, V_2 \dots V_n$  是赋给  $f_1, f_2 \dots f_n$  的逻辑值, 用集合  $J = \{f_1 = V_1, f_2 =$

$V_2 \dots f_n = V_n\}$  表示信号赋值。如果赋值  $J$  能使门  $g$  变为确认门, 则称这种赋值过程为确认。

未确认门指出在递归学习中需要赋值的位置, 而确认则决定对信号线赋什么值。

**定义 3**  $G_c = \{J_1, J_2, J_n\}$ , 其中  $J_i, i([1, n])$  为未确认门  $g$  的确认。如果对于任意确认  $J^*$ , 总有  $J^* \subseteq J_i, J_i \subseteq G_c$  则称确认集合  $G_c$  是完全的。

确认的完全性是保证正确蕴含的基础。仍以与门  $g$  为例, 当输出赋值为 0 时, 门  $g$  为未确认门。某个确认集  $C = \{J_1, J_2\}, J_1 = \{a=0\}, J_2 = \{b=0\}$ 。其中  $a, b$  为门  $g$  的输入。确认  $J^* = \{a=1, b=0\}$  没有包含在集合  $C$  中, 但由于  $J^* \subseteq J_2$ , 类似可知确认集  $C = \{J_1, J_2\}$  是完全的。

**定义 4** 对于给定电路  $C$  和电路上部分信号线的赋值,  $f$  为电路中任意一个未赋值的信号线。如果在所有相互一致的赋值作用下, 没有剩下未确认门, 并且一致包含有  $f = V, V$  是某逻辑值, 则在电路  $C$  的这种赋值条件下, 称赋值  $f = V$  是必须的。

用 NV (Necessary Value) 表示必须的赋值。如能找出所有必须的赋值时, 称蕴含是精确的。递归学习的算法的主要思想是尽可能地找出数字硬件逻辑图中, 各个逻辑端口之间的逻辑约束关系 (指直接和间接的逻辑蕴涵关系)。这就可以尽可能地减少寻找测试向量过程中错误判定的可能性, 甚至完全避免错误判定, 从而避免回溯。递

归学习的算法流程如下:

```

initial r=0
/* r 为递归层数。 r-max 为设定最大递归层数 */
Recursive-learnings(r, r-max)
{ //作直接蕴含
direct-implication(unjustified gates);
//根据蕴含的到未确认门集合
setup-unjustified-set();
if(r< r-max)
{
for(each gate in unjustified set)
{ //建立门 g 的确认集合
setup-justification-set();
for(each justification in justification set)
{
Recursive-learnings(r+1, r-max);
}
}
if(for all justifications, f=v)
{
//用 f=v 作直接蕴含
direct-implication(f=v);
}
}
}
}

```

这个学习过程会一直进行直到遇到递归层次限制,通常这个数越大,学习的逻辑约束关系越多,后面的判定过程也越快,但同时学习的时间也越长,根据实验中的表现,我们认为 3—5 层是比较合适的。该算法具有下面几个特点:

- (1) 一种完备的算法,具有一般性;
- (2) 该算法不仅限于三值逻辑系统,可以应用于五值系统、七值系统等;
- (3) 算法中定义了一个最大递归深度  $r\text{-max}$ , 算法的时间复杂度是  $r\text{-max}$  的指数,但其空间复杂度则随着  $r\text{-max}$  呈线性增长。但最大递归深度的确定是一个 NP 问题;
- (4) 不仅可以单独使用,还在别的算法中,利用递归学习算法提高算法效率;
- (5) 可以用递归学习算法对电路进行逻辑优化、逻辑综合。

完成学习过程后,就要进行判定过程,即判定某个门的扇出是否能为某个逻辑值。在等价性检验中,要判定 miter 的输出为 1,即最后一个逻辑门“XOR”的扇出为 1,如果成立,表示组成 miter 的两个电路不等价,否则才等价。

判定过程先是对要判定的逻辑门的扇出进行逻辑信号赋值,然后判断这个赋值是否可行,由于没有递归层次限制,这个逻辑信号可以尽可能远的传播。如果遇到矛盾,表示这个电路对于任何输入,该逻辑门的扇出不可能为其指定的值,否则就表示其扇出的该逻辑值可以出现。通过这种方法,就能判定某个逻辑门是否有固定的扇出,即解决可满足性问题。

## 4 实验结果

实验结果是在 Pentium IV cpu 1.8GHz、512MB 内存的 PC 机上实现的,实验对 ISCAS' 85 benchmark 的部分电路实现了两种程序:BDD 方法和递归学习方法,计算时间如表 1 所示。其中,BDD 表示单独采用 BDD 进行等价性检验,RL 表示利用递归学习方法进行等价性检验。从表 1 中可以看出利用递归学习的方法有效地提高了算法的效率。

表 1 计算时间比较

电路名	原始输入	原始输出数	门数	BDD( s)	RL ( s)
C432, C432nr	36	7	316	1. 32	0. 8
C499, C499nr	41	32	403	12. 45	5. 4
C1355, C1355nr	41	32	1091	15. 82	9. 1
C1908, C1908nr	33	25	1757	10. 46	48. 3

5 结束语

等价性检验是指对照设计规范来验证设计实现的功能正确性。等价性检验方法仅仅验证设计实现与对应的设计规范之间是否功能等价, 它不能保证设计规范本身的正确性。文章分析了等价性检验的一般方法; 讨论了基于递归学习的组合电路等价性检验方法。递归学习的方法利用直接蕴含和间接蕴含, 实验结果表明了该方法的可行性和有效性。在实际应用中, 通常结合其他算法进行递归学习, 从而有效地提高算法的效率。下一步的工作是研究更有效结合递归学习的方法, 从而提高算法的效率。

参考文献:

[ 1] Dr. Eng. Ka Lok Man. Efficient Equivalence Checking of Industrial Designs[ C] . In Proc. of IEEE Design Verification Conference in Europe. 2001: 1—10.

[ 2] Shi-Yu Huang, Kwang-Ting (Tim) Cheng, Formal Equivalence Checking and Design Debugging[ M]. Boston: Kluwer Academic Publishers. 1998.

[ 3] D. Brand. Verification of large synthesized designs[ C] . In Proceedings of International Conference of Computer-aided Design, 1993: 10—20.

[ 4] H. Fujiwara, T. Shiono. On the acceleration of test generation algorithms[ J] . IEEE Trans. on Computers, 1983, 32( 12) : 1137—1144.

[ 5] M. Schulz, E. Frischler, T M. Sarfert. SOCRA TES: A highly efficient automatic test pattern generation system[ J] . IEEE Transaction on Computer-Aided Design, 1987, 7( 1) : 126—137.

[ 6] Y. Matsunaga. An Efficient Equivalence Checker for Combinational Circuits[ C] . In 33rd Design Automation Conference, 1996: 629—634.

[ 7] 张镭, 林争辉, 吕宗伟. 递归学习寻找对称变量[ J] . 上海交通大学学报, 2002, 36( 12) : 1709—1712.

[ 8] 李晓维, 吕涛, 李光辉. 集成电路设计验证[ J] . 中国科学院计算技术研究所内部刊物——信息技术快报, 2004, ( 9) .

[ 9] 李光辉, 邵明, 李晓维. 基于 BDD 的组合电路等价性检验方法[ J] . 微电子学与计算机, 2003, ( 2) : 48—51.

[ 10] 李光辉, 邵明, 李晓维. 通用 CPU 设计验证中的等价性检验方法[ J] . 计算机辅助设计与图形学学报, 2005, 17( 2) : 230—235.

Combinational equivalence check based on recursive learning

ZENG Qiong<sup>1,2</sup>

( 1. Chengdu University of Information Technology, Chengdu 610225, China; 2. Chengdu Institute of Computer Application, Chinese Academy of Sciences, Chengdu 610041, China)

**Abstract:** The equivalence check usually includes both functional and structural verifications. A general equivalence check method is analyzed. The combinational equivalence check methods based on recursive learning algorithm are discussed. Both direct implication and indirect implication methods are used in this algorithm. Therefore the SAT is solved.

**Key words:** recursive learning; equivalence check; combinational circuit